Chapter 6

Statistical MWE-aware parsing

Mathieu Constant
ATILF UMR 7118, Université de Lorraine/CNRS

Gülşen Eryiğit
Istanbul Technical University

Carlos Ramisch
Aix-Marseille Université

Mike Rosner
University of Malta

Gerold Schneider
University of Konstanz and University of Zurich

This chapter aims at presenting different strategies that have been designed to incorporate multiword expression (MWE) identification in the process of syntactic parsing using statistical approaches. We discuss MWE representation in treebanks, pipeline and joint orchestrations, the integration of external lexicons and the evaluation of MWE-aware parsers, concluding with our suggestions for future research.

1 Introduction

Supervised STATISTICAL PARSING is nowadays an important and challenging field of natural language processing (NLP). It consists in predicting the most probable syntactic structure of a new sentence, given a statistical model that has been trained on a TREEBANK, that is, a syntactically annotated corpus. Since the seminal works of Nivre & Nilsson (2004) for dependency parsing and Arun & Keller

Mathieu Constant, Gülşen Eryiğit, Carlos Ramisch, Mike Rosner & Gerold Schneider. 2019. Statistical MWE-aware parsing. In Yannick Parmentier & Jakub Waszczuk (eds.), *Representation and parsing of multiword expressions: Current trends*, 147–182. Berlin: Language Science Press. DOI:10.5281/zenodo.2579043

(2005) for constituency parsing, a new research line has emerged: incorporating the analysis of multiword expressions (MWEs) in such parsers. The main objective of this chapter is to present different approaches that have been developed and evaluated for statistical MWE-aware parsing systems.

The design of MWE-aware parsers must address the following questions: How are MWEs represented in combination with syntactic trees? When is MWE identification performed with respect to parsing? What algorithms and machine learning techniques are to be used for the two tasks? How can external lexical resources be integrated to improve MWE coverage? How are systems evaluated?

Answering the question about MWE REPRESENTATION is fundamental as it enables the definition of a system's output. Hence, it influences the design of datasets used for training and testing, including treebanks, as shown in Section 3.

The ORCHESTRATION issue is also crucial in order to position MWE identification with respect to parsing: should it be performed before, during, or after it? The answer is not straightforward as it might depend on the type of MWE (Eryiğit et al. 2011). Orchestration also implies determining how the two components interact. For instance, in pipeline strategies (before or after) discussed in Section 4, should the intermediate input/output be computed using MWE concatenation strategies or MWE substitution ones? Joint strategies (during) discussed in Section 5 alongside *n*-best strategies, might involve different methods like adapting a grammatical formalism for constituency parsing (Green et al. 2013) or concatenating arc labels in dependency parsing (Vincze et al. 2013).

Concerning ALGORITHMS and machine learning, most techniques use workaround approaches by adapting the MWE-aware representation to existing representations directly exploitable by off-the-shelf tools (Nasr et al. 2015). Nonetheless, new parsing algorithms have been recently proposed that include specific handling of MWEs, notably when using joint strategies (Nivre 2014).

The integration of EXOGENOUS LEXICAL KNOWLEDGE in the system, discussed in Section 6, is non-trivial but potentially helpful. Indeed, supervised systems are trained on datasets of limited size. Therefore, one drawback of such systems is the limited coverage in terms of MWEs. One possible solution consists in integrating knowledge coming from large-scale MWE lexicons, either manually built and/or validated (Candito & Constant 2014) or automatically acquired (Schneider 2012).

The last issue concerns EVALUATION: what is the impact of MWE identification on syntactic parsing and vice-versa? What types of measure are adequate to quantify this impact? We try to answer these questions in Section 7.

The outline of this chapter is as follows. First, we briefly explain some basic concepts and terms in statistical parsing in Section 2. Then, each section addresses the questions above. We conclude in Section 8 by providing a summary of the current research in statistical MWE-aware parsing and presenting pointers that, in our opinion, may lead to significant advances in the field in the future.

2 Statistical parsing

Parsing, also referred as syntactic analysis, is the process of assigning a syntactic structure to a given input sentence. The analysis is aimed at producing a valid syntactic tree conforming to a hand-written or automatically induced language grammar. With the emergence of manually annotated datasets (i.e. treebanks) and machine learning techniques, statistical parsing (Collins 1996; Charniak 2000) has become the dominant approach in the parsing literature.

Statistical parsing aims at selecting the most probable parse tree from the set of all possible parse trees for a given sentence. These data-driven parsing models may be basically grouped under generative or discriminative approaches. Generative parsing models generally rely on a grammatical formalism whereas discriminative ones are usually performed without any underlying grammar. There exist also joint approaches where a discriminative model is used to rerank the top *n* candidates of a generative parser.

Constituency and dependency formalisms are the two most common parsing formalisms used in statistical parsing. Figure 1 and Figure 4 each provide constituency and dependency parse tree samples for the sentence *The prime minister made a few good decisions.*

In the constituency formalism, a sentence is regarded as being composed of phrases and parsing is the task of determining the underlying phrase structure. For example, a statistical generative constituency parser aims to assign probabilities to a parse tree by combining the probabilities of each of its sub-phrases. In the dependency formalism, parsing is defined as correctly determining the dependency relations between words of an input sentence. More precisely, the aim of dependency parsing is to correctly determine the dependent-head relationships between words and also the type of these relationships such as subject, object, predicate. Dependency parsing is nowadays strikingly more popular than constituency parsing and attracts the attention of an ever-growing community in NLP. Furthermore, most existing MWE-aware parsers are developed in the dependency framework. Therefore, in this chapter, we focus mainly on different orchestration scenarios applied for different statistical dependency parsing approaches.

The two commonly used approaches for statistical dependency parsing in the literature are transition-based (Yamada & Matsumoto 2003; Nivre et al. 2007) and graph-based (Eisner 1996; McDonald et al. 2006; Nakagawa 2007). Transition-based approaches treat the dependency parsing task as the determination of parsing actions (such as push/pop operations in a shift-reduce parser) by the use of a machine learning classifier. Graph-based approaches treat parsing as finding the most likely path within a graph, such as the highest-scoring directed spanning tree in a complete graph. Most MWE-aware parsing strategies are adaptations of standard parsers experimenting with various models of orchestration concerning the scheduling of MWE identification with respect to syntactic analysis.

MWEs pose *challenges* for all areas of NLP, and statistical parsing is not an exception. An MWE may be *ambiguous* among accidental co-occurrence, literal, and idiomatic uses. The possible surface forms of an MWE *vary*, especially due to morphological variations which may become radical in morphologically rich languages. MWE components do not have to appear in consecutive locations within a sentence and it is hard to correctly identify a *discontinuous* MWE by ignoring the intervening words. The syntactic *non-compositionality* of MWEs may result in irregular parse trees. The ambiguous, discontinuous, non-compositional and variable nature of MWEs needs to be carefully handled during parsing in order to produce a valid syntactic structure. Additionally, annotated datasets (treebanks) are crucial resources for the training of data-driven statistical parsers. The *scarcity and limited size* of MWE-annotated treebanks is a great challenge faced by MWE-aware parsing.

3 MWE representations in treebanks

The choice of an appropriate MWE representation is crucial, with strong consequences on the format of treebanks. Representational choices that have affected existing treebanks in this way range from words-with-spaces – e.g., the French treebank (Candito & Crabbé 2009) – to the use of special MWE syntactic relations – e.g., the Universal Dependencies project (Nivre et al. 2016). Some treebanks may not even contain MWE representations at all, while others may have sophisticated multi-layer representations (Bejček et al. 2012).

The number and variety of available MWE-aware treebanks is growing (Rosén et al. 2015). They do not necessarily cover the same kinds of MWEs. They often belong to the constituency or the dependency frameworks, but some can also be compatible with different types of grammatical formalisms, like lexical functional grammar (Dyvik et al. 2016). To narrow down the scope of this section, we focus on MWE representations in relation to treebanks *that are useful to or that have been used in statistical MWE-aware parsing*.

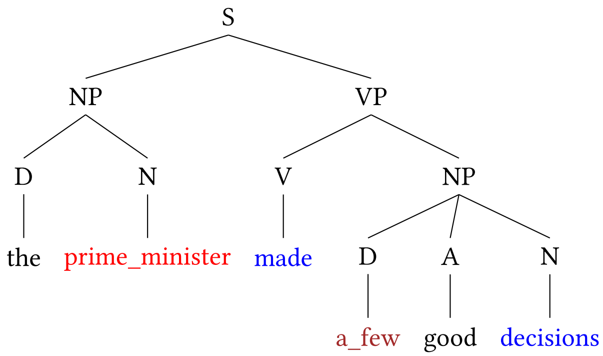

Figure 1: Constituency MWE-aware tree with words-with-spaces representation

3.1 No representation at all

The simplest and most obvious MWE representation is not to consider MWEs at all, only considering separate word tokens. While such a treatment is simplistic, it also has a number of advantages. First and foremost, it is easy to operationalize: no distinction is necessary between single words in combination and MWEs. MWEs include a variety of phenomena: compound nouns, technical terms, multiword entities, light-verb constructions, phrasal verbs, idioms, and proverbs. In general they are partly non-compositional, but due to this characteristic they also border on or overlap with collocations, which are an inherently gradient phenomenon. Not representing MWEs can thus be seen as a tacit assumption that all forms of MWEs are gradient.

Statistical parsers were conceived to improve parsing performance by modeling lexical interactions (Gross 1984; Sinclair 1991; Collins 1999). As MWEs are a subclass of collocations, the statistical attraction between the participating words is typically very strong and errors are therefore much rarer. Statistical parsers generally perform better on relations that are semantically expected (as e.g., in selectional preferences), so performance on verb complements for example is much higher than on verb adjuncts.

3.2 Words-with-spaces representation

A simple representation consists in considering MWEs as single nodes of the syntactic tree (Sag et al. 2002), such as in the strategy adopted in the LFG/XLE parser described by Angelov (2019 [this volume]). This “words-with-spaces” representation implies that MWEs have an atomic interpretation. In the constituency framework, the MWE forms are leafs. Their parent nodes correspond to their parts-of-speech (POS) category, as shown in Figure 1. For instance, *prime minister* has a noun parent node and *a few* has a determiner parent node. A concrete example where MWEs are represented this way is the first version of the French treebank distributed for parsing (Candito & Crabbé 2009). In the dependency framework, the MWE node has the same linguistic attributes as a single word token: POS tag, lemma and morphological features. For instance, *hot dogs* would be a noun in plural, whose lemma is *hot dog*. Such representations imply that MWEs have been pre-identified and represented as word-with-space tokens before parsing. Moreover, they have several drawbacks in terms of linguistic expressiveness. First, discontinuous MWEs like the light-verb construction *make decisions* in Figure 1 cannot be represented this way. Then, the semantic processing of semi-compositional MWEs might be problematic as the internal syntactic structure is impossible to retrieve.

3.3 Chunking representations

Another way of representing MWEs uses CHUNKING. Chunks are a polysemous concept, but its two meanings are related. On the one hand, chunks are seen as psycholinguistic units that are partly or fully lexicalized, that is, stored as one entity in the mental lexicon (Miller 1956; Pawley & Syder 1983; Tomasello 1998; Wray 2008). On the other hand, they are the concrete output of applying finite-state technology to obtain base-NPs and verb groups deterministically. While the psycholinguistic and the computational concepts are related, the latter has the drawback that chunks need to be continuous.

Black et al. (1991) pointed out that dependency grammars are particularly suited to model chunks and parse between heads of chunks. In fact, chunks are close to Tesnière's original conception of nucleus, which is typically not a single word (Tesnière 1959). Some dependency parsers following this scheme exist, for example Schneider (2008). Nivre (2014) has proposed a transition-based parser that performs MWE merging as it syntactically parses a sentence. This operation can be seen as MWE chunking.

A standard way of representing chunks in tagging systems is the IOB annotation scheme (Ramshaw & Marcus 1995).[1] Such representations have been successfully adapted to named entity recognition (Tjong Kim Sang 2002) and MWE identification (Vincze et al. 2011; Constant et al. 2012). For MWEs, there are variants covering continuous MWEs (Blunsom & Baldwin 2006) and gappy

[1] Tokens are tagged as "B" for *begin*, "I" for *inside* and "O" for *outside* a chunk.

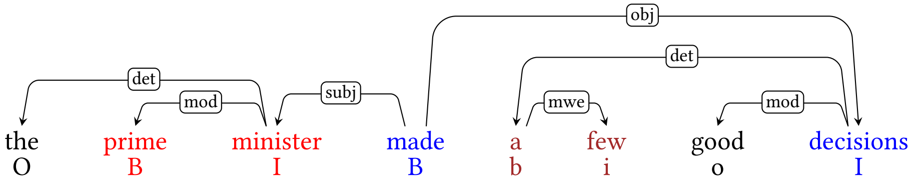

Figure 2: Chunking-based representation with IOB tags (Schneider et al. 2014)

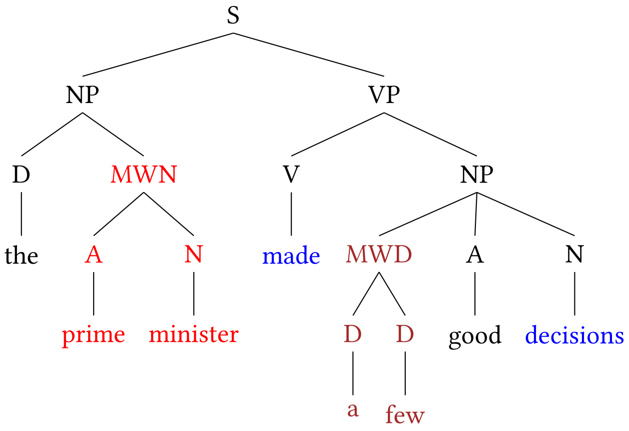

Figure 3: Flat constituency subtree representation (Green et al. 2011)

ones (Schneider et al. 2014). For instance, Schneider et al. (2014) use a 6-tag set (with additional lowercased tags in order to emphasize nested MWE structures) to represent MWEs enabling 1-level nesting, as shown in Figure 2. Such representations can be used in treebanks for training pipeline MWE-aware systems (Section 4) and joint MWE-aware parsers (Section 5).

3.4 Subtree representations

Another way of representing MWEs is to annotate them as SUBTREES made of several nodes of the syntactic tree. Many treebanks using such representations can be found in Rosén et al. (2015). Several types of subtree MWE representations were proposed in treebanks, according to the language, MWE type and syntactic formalism.

For processing purposes, words-with-spaces representations have often been automatically converted into flat subtrees. In the constituency framework, an

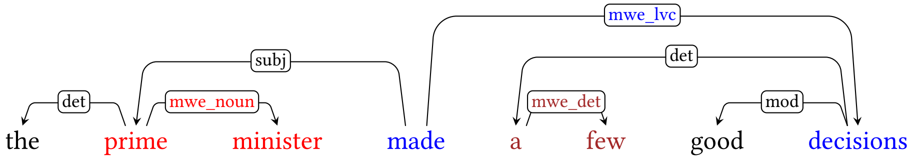

Figure 4: Flat head-initial dependency subtree representation

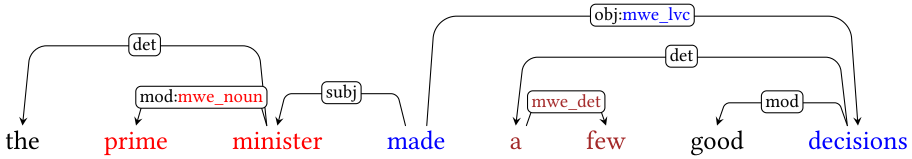

Figure 5: Structured dependency subtree representation with extended labels

MWE is considered as a special constituent with a given POS tag. MWE components are leaves of the MWE subtree, as shown in Figure 3.[2] There exist different variants for constituency treebanks (Głowińska & Przepiórkowski 2010). This representation has been used by Arun & Keller (2005) and Green et al. (2011), especially for compounds. In the dependency framework, flat subtrees can be either head-initial, that is, the root of the subtree is the first token (Nivre et al. 2004; Seddah et al. 2013), or head-final, with the root being the last token of the MWE (Eryiğit et al. 2011). All other MWE component tokens depend on this arbitrarily defined head, as shown in Figure 4. This representation is used, for example, in the Universal Dependencies treebanks (Nivre et al. 2016).

Flat subtree representations have a disadvantage: the internal syntactic structure of MWEs, required for semi-fixed MWEs in particular, is lost, like for words-with-spaces representation. To retain the internal syntactic structure as well as the MWE status, some authors propose representing an MWE with its syntactic subtree, where arc labels are extended with MWE tags, as shown in Figure 5. This kind of representation has been used, for instance, for annotating light-verb constructions (Vincze et al. 2013) and continuous MWEs (Candito & Constant 2014).

Candito & Constant (2014) adopt a hybrid representation scheme to distinguish regular from irregular MWEs. Regular MWEs have a regular syntactic

[2] MWE-related symbols MWN and MWD respectively stand for *multiword noun* and *determiner*.

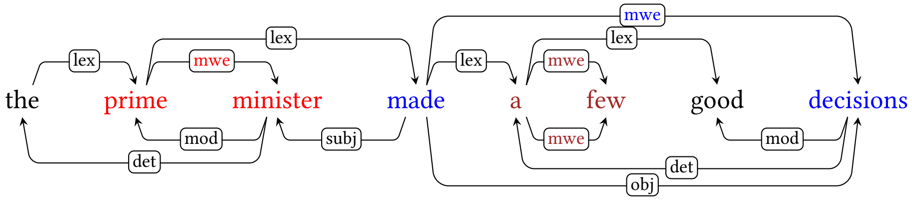

Figure 6: Representation on two distinct layers (Constant et al. 2016)

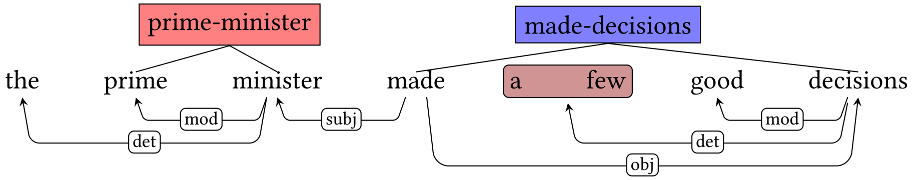

Figure 7: Representation on factorized lexical and syntactic layers (Constant & Nivre 2016)

structure[3] whereas they display semantic irregularity. They are represented with structured MWE subtrees, as in Figure 5. Irregular MWEs display an irregular syntactic structure (e.g., *by and large* is the coordination of a preposition and an adjective) and therefore cannot be analysed syntactically in a compositional way. They are represented with flat subtrees, as in Figure 4.

3.5 Multilayer representations

One of the most interesting MWE representations combined with (deep) syntactic analysis is the one used in the Prague Dependency Treebank (Bejček et al. 2012). It combines three different analysis layers in the form of trees: morphological (*m*-layer), syntactic (*a*-layer) and "semantic" ones (*t*-layer). Nodes of one layer can be linked to nodes of another layer to model the interleaving of the different types of analysis. MWEs are represented on the t-layer and are associated with MWE entries of a lexicon. To our knowledge, there is unfortunately no statistical parser outputting such combined structures.

Though less linguistically expressive, other multilayer representations have been proposed on top of a combined lexical and syntactic parser. The proposal of Constant et al. (2016) is to have two distinct layers for representing lexical and syntactic analysis in the form of dependency trees. The two layers share the same

[3] The distinction between irregular and regular MWEs is arbitrary, being defined by a manually-built set of POS patterns.

nodes, that correspond to the tokens, as shown in Figure 6. The syntactic layer represents the syntactic structure in the dependency framework. The lexical layer represents the lexical segmentation in the form of a tree. Arcs in MWE subtrees have a special label "mwe". For instance, the MWE *prime minister* corresponds to a subtree whose root is *prime* and which is composed of an "mwe" arc from *prime* to *minister*. In order to form a unique tree for the lexical layer, lexical units are sequentially related via arcs labeled "lex". For instance, the MWE *prime minister* is linked to the following lexical unit *made decisions*. This dual representation has several advantages. First, syntactic and lexical analyses are explicitly separated. In the case of regular MWEs, there is a clear distinction between the syntactic and the semantic status (regular syntactic structure vs. irregular semantics). In addition, the representation enables not only nested MWEs to be annotated (e.g., *a few* in *made a few good decisions*) but also fully overlapping expressions (e.g., the noun compound *rain check* inside the light verb construction *to take a rain check*). On the down side, irregular MWEs are duplicated on the two layers because there is no possible compositional syntactic analysis (e.g., *a few*). Additionally, arcs linking lexical units could be made implicit, as they can straightforwardly be computed from their positions in the sequence.

Constant & Nivre (2016) correct the main drawbacks of the previous two-layer representation by making it more compact and more factorized. The representation is still composed of two layers, but the lexical layer is a forest of constituent-like trees representing complex lexical units like MWEs, as shown in Figure 7. Here, the discontinuous MWE *made decisions* is represented by a tree whose root corresponds to a new lexical node having linguistic attributes like any token: a form (*made decisions*), a lemma (*make decision*), a POS tag (verb) and morphological features (past tense). It is straightforward to elegantly represent embedded and fully overlapping MWEs, as lexical units are trees. Irregular MWEs like *a few* and simple words are called SYNTACTIC NODES. The syntactic layer is a dependency tree over such nodes. Therefore, irregular MWE nodes and simple word nodes are shared by the two layers. For example, there is a "det" arc from *decisions* to *a few*, as it is compositionally modified by the complex determiner. This representation is not without some limitations: the lexical layer cannot represent an MWE that strictly requires a graph (and not a tree). For instance, it is impossible to represent the coordinated MWEs *had shower* and *had bath* in the sentence *John* $\textit{had}_{1,2}$ *a* $\textit{shower}_{1}$ *then a* $\textit{bath}_{2}$.

4 Pipeline approaches

A minimal processing pipeline consists of a collection of two processes arranged in a chain so that the output of the first process is the input of the other. Thus, a processing pipeline for statistical MWE parsing involves two processes, one to identify the MWEs in the input sentence, and another for parsing the sentence into one or more structures that include the MWEs. The question that we address in this section concerns the order in which these two processes are arranged, and there are clearly two possibilities referred to as preprocessing (Section 4.1), and postprocessing (Section 4.2).

4.1 Preprocessing approaches

Preprocessing means that the MWE identification task takes place before parsing. For the parser to benefit from this, a decision must be made about how to represent MWEs in the input. As discussed earlier, there are different approaches, the most important of which employ concatenation (Section 4.1.1), or substitution (Section 4.1.2) operations, as discussed in the following sections.

4.1.1 Concatenation approach

A widely used pipeline approach to statistical MWE-aware parsing is to have a RETOKENIZATION phase before parsing. It consists in first pre-identifying MWEs, then concatenating their components in one single token, and finally applying a syntactic parser trained on a treebank where MWEs have a words-with-spaces representation (Section 3.2). Note that this approach is limited to continuous MWEs.

For example, given the input token sequence *The prime minister made a few good decisions*, the MWEs *prime minister* and *a few* are first pre-identified. Each of them is then merged by concatenating its components into a single token. The sequence is retokenized as *The prime_minister made a_few good decisions* and is then parsed. This approach has the advantage of reducing the token-count of the sentence and hence reducing the search space of the parser. However, it may not be realistic to recognize some types of MWEs without access to morpho-syntactic information.

Seminal studies on gold MWE identification performed before either constituency parsing (Arun & Keller 2005) or dependency parsing (Nivre et al. 2004; Eryiğit et al. 2011) showed that it may have a great impact on parsing accuracy. Other studies confirmed that more realistic MWE pre-identification actually helps parsing. Korkontzelos & Manandhar (2010) evaluated MWE pre-identification using Wordnet 3.0 for lexicon lookup before shallow parsing. The set of MWEs was limited to two-word continuous compound nominals, proper names, and adjective-noun constructions. The authors showed that the approach improves shallow parsing accuracy. For instance, without MWE pre-identification, *he threw the fire wheel up into the air* is erroneously parsed as: *(he) (threw) (the fire) (wheel up) (into) (the air)*, whereas with MWE pre-identification the result is: *(he) (threw) (the fire_wheel) (up) (into) (the air)*. Cafferkey et al. (2007) carried out similar experiments with a probabilistic constituency parser. MWEs were automatically identified by applying a named entity recognizer and list of prepositional MWEs. A slight but statistically significant improvement was observed. We should note that in the above studies, MWE identification itself was not evaluated.

The SPMRL shared task (Seddah et al. 2013) had a special track dedicated to MWE-aware parsing in French. The provided treebank included continuous MWE annotations represented as flat subtrees (Figure 4). All but one competing team did not develop special treatments for MWEs. The winning team was the only one to have a preprocessing stage to identify MWEs using a tagger based on linear conditional random fields (Constant, Candito, et al. 2013). The tagger model also incorporated features based on an MWE lexicon (Section 6.3).

4.1.2 Substitution approach

Another approach is to use substitution: whenever an MWE from the lexicon matches, it is replaced by its head word. Such approach is employed by Weeds et al. (2007) for technical terms (Section 6.2), and by Schneider (2008) on all chunks. In a typical substitution approach, for example, the term *natural language processing* would be replaced by *processing* before parsing.

The advantage of keeping the lexical head is that resources taking lexical relations into account, such as bi-lexical disambiguation (Collins 1999), can use the lexical information. Thus, potential sparsity problems are reduced in comparison to the concatenation approach. For example, the prepositional phrase attachment ambiguity in *We help users with natural language processing* can be resolved properly, even if *natural language processing* is unseen in the training data. As long as *processing* exists in the training corpus, the ambiguity can be solved because the combination *help-with-processing* is more likely than *user-with-processing*.

The potential drawbacks of this approach are that, on the one hand, strings may be ambiguous, and on the other hand non-compositionality may affect the results. Ambiguous strings are illustrated below: while the first sentence of each example is an MWE, the second is accidental cooccurrence. The last example involves light verbs, for which Tu & Roth (2011) use token-wise disambiguation, as ambiguity is relatively frequent.

(1) a. I saw her, and *by the way* she went there on foot.
    b. I recognized her *by the way* she walks.

(2) a. In *natural language processing*, humans are also challenged.
    b. In *natural language processing* can be difficult.

(3) a. The politician *took* a strong *position* on the issue.
    b. The soldier *took* a vanguard *position* on the mountain top.

Non-compositionality may lead to situations in which the head is semantically so different that attachment preferences are also affected.

(4) a. I saw the road with the *torch light*.
    b. I saw the road with the *traffic light*.

If the MWE *traffic light* is reduced to *light*, the chances are that the prepositional phrase is erroneously attached to the verb, as *see-with-light* is likely. If *traffic light* is treated as an MWE, bi-lexical disambiguation can only profit if very large annotated resources exist. Unless a backoff method to treat MWE components is included, the increased data sparseness may easily lead to worse results.

4.2 Postprocessing approach

In this section, we present approaches where parsing precedes MWE processing. We make a distinction between MWE identification and discovery. We define IDENTIFICATION as the process of recognizing MWEs in context, that is, as tokens inside running text. On the other hand, DISCOVERY aims at creating a lexicon of MWE types from the corpus. This lexicon can later be used to guide MWE identification and parsing. In this section, we describe approaches for identification after parsing (Section 4.2.1) and for discovery after parsing (Section 4.2.2), focusing on works in which the result of discovery was later employed for identification.

4.2.1 Post-parsing MWE identification

Identifying MWEs after syntactic parsing is a natural approach to MWE-aware parsing as an MWE generally constitutes a syntactic constituent. In the dependency framework, there is usually a path continuously linking the MWE components in the syntactic tree. As a consequence, pre-parsing is particularly relevant for detecting discontinuous MWEs, that is, MWEs that include alien elements, by employing adapted lexicon lookup methods. In Figure 7, the MWE *made decisions* is discontinuous. As there is an object arc from *made* to *decisions*, the two words are *syntactically* adjacent. A matching procedure taking the syntactic structure into account can therefore be beneficial for MWE identification. Furthermore, MWEs can have different syntactic variants. For instance, *a **decision** was **made** by John* is the passive voice variant of *John made a decision.* The detection of such syntactic variants obviously benefits from the result of syntactic parsing.

Fazly et al. (2009) identify verb-noun expressions in a parsed text based on a list of 60 candidate expressions. First, they identify candidate occurrences of the expressions using rules based on syntactic annotations and lexical values. Then, they discriminate MWEs from literal expressions using different methods. One is based on the assumption that a verbal MWE expression has *fewer syntactic variants* than its literal counterparts, giving rise to the heuristic that canonical forms are idiomatic (e.g., *pull one's weight*) and non-canonical variants are literal (e.g., *pull a weight, pull the weights*). Another method compared the distributional contexts of co-occurring verb-object pairs to two sets of gold-standard contexts: one for idiomatic readings and another one for literal readings.

Nagy T. & Vincze (2014) compare the use of parsers and of a syntax-based pipeline approach to identify verb particle constructions in English. English off-the-shelf parsers usually have a specific syntactic arc label to identify occurrences of verb-particle constructions. Nonetheless, such parsers tend to get good precision but low recall, as they do not use dedicated features for this task. The pipeline method developed in this paper uses a standard parser to identify a first set of candidates. This set is subsequently enlarged using other syntactic relations. A classifier is then applied in order to decide whether they are verb-particle constructions or not. They show a significant gain in terms of recall and F-score with respect to standard parsers on the Wiki50 corpus (Vincze et al. 2011).

4.2.2 Post-parsing MWE discovery

This section discusses the discovery of new MWEs after parsing. This is particularly useful for the creation of resources that can be used for MWE-aware parsing (Section 6). For instance, such lexicon of newly discovered MWEs can be subsequently used for MWE pre-identification at the next cycle of processing. Seretan (2011) has shown that discovery based on parsed corpora provides considerably cleaner results than those relying on shallow analysis (e.g., POS-tagged corpora). Foufi et al. (2019 [this volume]) discuss the integration of resources built with the help of MWE discovery into a language-independent symbolic parser.

Since the literature in MWE discovery is huge, we focus on two studies that represent a sample of this type of approach. Lehmann & Schneider (2011) and Ronan & Schneider (2015) used automatically parsed data for discovering MWEs of different types, including idiomatic verb + prepositional phrase (PP) combinations and light-verb constructions in English. These cases involved the use of different collocation extraction scores.

For discovering Verb-PP idioms the O/E score was used, combined with filters including T-score and Yule's K (which estimates the degree of non-modifiability of a candidate). Table 1 reproduces the results of discovery, sorting the candidates by descending O/E score. Among the top-ranked candidates, many are genuine idioms (e.g., *to kill two birds with one stone*).

Table 1: Top-ranked verb-object + preposition-noun tuples, using the the O/E score (Lehmann & Schneider 2011)

verb	object	prep	desc. noun	T-score	O/E
send	*shiver*	*down*	*spine*	5.74456	2.21477×10^{8}
tap	*esc*	*for*	*escape*	6.40312	2.1134×10^{8}
separate	*shield*	*from*	*plate*	6.78233	2.33384×10^{7}
refer	*gentleman*	*to*	*reply*	8.24621	7.8143×10^{6}
obtain	*property*	*by*	*deception*	5.2915	7.60043×10^{6}
ask	*secretary*	*for*	*affairs*	6.40312	5.01529×10^{6}
kill	*bird*	*with*	*stone*	5.38516	3.37917×10^{6}
add	*insult*	*to*	*injury*	6.08276	2.21769×10^{6}
throw	*caution*	*to*	*wind*	5.09902	2.03157×10^{6}
refer	*friend*	*to*	*reply*	7.54983	1.36298×10^{6}
report	*loss*	*on*	*turnover*	7.14142	1.34742×10^{6}

For discovering light-verb constructions, the t-score was used together with a number of filters including WordNet and NomBank lookup (Ronan & Schneider 2015). An example of analysis is shown in Figure 8, showing a precision and recall plot by candidate list length. The vertical axis shows precision and recall, respectively, the horizontal axis (which is logarithmic) gives the cutoff in the ranked list of candidates to be included in the evaluation. For the cutoff at 20, the reported candidates for *give*+object, precision is 100%, while recall is 10%. At rank 2560, about 88% of all instances in the gold standard were found.

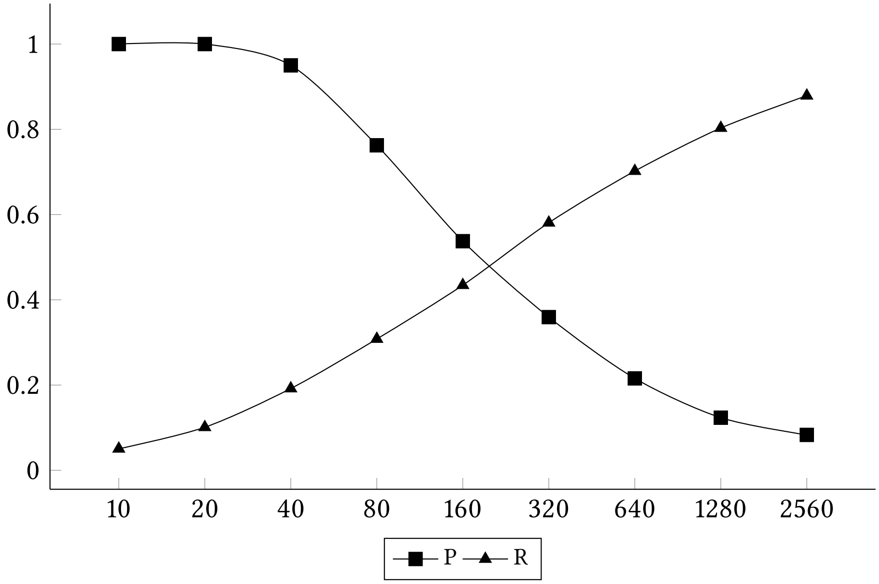

Figure 8: Precision vs. recall curve of the light verb *give* in the British National Corpus, using t-score (Ronan & Schneider 2015)

5 Joint approaches

Joint approaches perform parsing and MWE identification simultaneously. Since syntactic and lexical-semantic information are complementary, both processes can help each other if performed together. In such systems, MWE lexical-semantic segmentation is often seen as a by-product of syntactic analysis, or vice-versa.

Some MWEs require quite sophisticated syntactic information to be recognized, such as subcategorization frames and phrase structure. Joint approaches favor delaying the decision as to whether a given combination is an MWE to the parser, where this information is available. In other words, the system has access to the right information at the right moment.

Parser evaluation scores are often reported on standard test sets, where MWEs have been manually pre-identified (gold). Jointly performing MWE identification and parsing is more realistic than parsing pre-annotated test sets, where MWEs are often represented as words with spaces (Figure 1). Indeed, when moving from standard test sets to real texts, gold MWE identification is not necessarily available. It may be hard to use a pipeline approach (Section 4) if the target MWEs are ambiguous or discontinuous.

On the downside, parsers that perform both syntactic analysis and MWE identification simultaneously are harder to design. First, ambiguity is increased, often by a larger number of labels and/or parsing decisions that are possible at a given moment. It is crucial, for such systems, to have coherent MWE annotations in treebanks, datasets that are large enough, and features that generalize well.

We classify such approaches according to the degree of "MWE-awareness" of the parser. In shallow approaches, the parser generates *n*-best solutions without putting any particular emphasis on MWEs, then uses MWE information for reranking (Section 5.1). The majority of joint approaches add MWE information to training and test treebanks, and then use off-the-shelf parsers enriched with dedicated MWE features (Section 5.2). We also present fully MWE-aware parsers that take them into account in the parsing algorithm itself (Section 5.3).

5.1 *n*-best and reranking approaches

One possible orchestration solution is to consider MWE identification as a retokenization problem, as described in Section 4.1.1. In *n*-best approaches, however, the text is first segmented into tokens in a non-deterministic way, considering *several possible segmentations*. Usually, the output of such non-deterministic tokenizer is a lattice containing all possible segmentation paths for a sentence (Sagot & Boullier 2005). This representation is particularly suited for ambiguous irregular constructions, that could be considered as MWEs or as accidental co-occurrence, depending on the context. The parser then must take this ambiguous segmentation and uses simple parsing models to disambiguate the input and generate a parse tree (Nasr et al. 2011).

An *n*-best MWE identifier is used by Constant, Le Roux & Sigogne (2013), producing a lattice of possible segmentations. Then, a PCFG-LA parser is used to disambiguate the possible readings. The authors test two variants. First, they consider that MWEs in the lattice are single nodes (words with spaces). Thus, different segmentation possibilities in the lattice are represented by paths with different lengths. Second, they consider that MWE components are individual nodes tagged using an IOB scheme, like in Figure 2. The latter obtains better performance because all possible paths in the lattice have the same length, resulting in more accurate parsing scores.

Conversely, the parser can use the same kind of approach and also generate *n*-best parsing trees. A reranker can then use MWE-aware features, among others, to choose the highest scoring tree. Constant et al. (2012), for instance, use a deterministic tokenizer but output *n*-best MWE-aware syntactic trees using the Berkeley constituency parser. Then, they use a discriminative reranker to choose the correct parse tree that includes MWE features.

These are considered joint approaches because, even though MWE segmentation and parsing are independent processes, one needs to be aware of the format of the input/output of the other. For example, the parser has to be able to process lattices as input, provided by the non-deterministic MWE identifier.

5.2 Treebank modification approaches

In Section 3, we discussed several ways to represent MWEs in treebanks. Standard statistical parsers trained on such treebanks will be inherently aware of MWEs, provided that they can handle the particular MWE representation in that treebank. For example, if MWEs are represented as subtrees (Figure 5), then there is no need to explicitly handle MWEs (Nivre et al. 2016). This subsection covers MWE-aware parsing studies in which the learning and parsing algorithms *remain unchanged* with respect to their standard version.

Approaches discussed in this section face several challenges. First, most of the time MWEs are either *absent* from treebanks, or the available representation requires *adaptations* in order to be usable by the parser. Second, parsers learned from MWE-annotated treebanks often require *extra features* to take MWEs into account properly. Third, these features may suffer from data *sparseness*, as individual MWEs may not occur often enough in limited-size treebanks.[4]

In this subsection, we present approaches that tackle the challenges posed by MWEs by:

- adding or modifying the MWE representation in the treebanks, and/or
- adding MWE-dedicated features to the parsing model.

The last challenge, related to data sparseness and domain adaptation, is tackled by integrating external resources in the parser, as discussed in Section 6.

In constituency parsing, several parsers, MWE representations and feature sets have been tested, especially on continuous MWEs in the French treebank. Constant, Le Roux & Sigogne (2013) experiment with two implementations of a PCFG-LA parser, using a representation similar to the one of Green et al. (2011) and a variant similar to IOB encoding.

When MWE annotation is absent, a reasonably straightforward solution is to automatically project an MWE lexicon on the treebank before training the parser. For instance, Kato et al. (2016) project a lexicon of compound function words (e.g., *a number of*) onto the English Ontonotes constituency treebank. Syntactic trees

[4] Some MWE categories may never occur (e.g., colloquial idioms) because many existing treebanks cover a single register (e.g., newspapers).

are modified to take MWEs into account. Constituents are then automatically transformed into dependencies and a standard first-order graph-based parser is learned. While the training data is modified, no MWE features are added to the model.

Early experiments on MWE-aware dependency parsing compared two representation variants: MWEs as subtrees or as words with spaces (Nivre & Nilsson 2004). The results indicated that the subtree representation (joint approach) is worse than parsing MWEs as words with spaces (pipeline approach). However, these results were obtained assuming gold MWE segmentation.

Vincze et al. (2013) were among the first to use a dependency parser to perform realistic MWE identification. They focus on light-verb constructions (LVCs) in Hungarian. They first perform an automatic matching of two annotation layers in the Szeged treebank: syntactic dependencies and LVCs. As a result, the dependency link between a light verb and a predicative noun (e.g., OBJ) is suffixed with a LVC tag, whereas regular verb-argument links remain unchanged, like in Figure 5. An off-the-shelf parser is used to predict the syntactic structure of sentences, including LVC links. Given that Hungarian is a relatively free word-order language, LVCs often involve long-distance dependencies. When compared with a classifier baseline, the parser performs slightly worse on continuous LVC instances (F1 = 81% vs. 82.8%) but considerably better on discontinuous LVCs (F1 = 64% vs. 60%).

Treebanks containing MWEs as words with spaces pose problems when converted into subtrees. When splitting an MWE, one needs to manually or semi-automatically assign POS tags, lemmas and morphological features to the individual MWE components. Additionally, the internal syntactic structure must be inferred. Since it is difficult to automate this task, the internal syntactic structure of decomposed MWEs is often underspecified using flat head-initial subtrees (Seddah et al. 2013), head-initial (Nivre et al. 2016) or head-final chained subtrees (Eryiğit et al. 2011), as detailed in Section 3.4. Eryiğit et al. (2011) compare parsing and MWE identification accuracy on different treebank representations for different MWE types. Their original treebank includes MWEs as words with spaces, which are semi-automatically transformed into subtrees. Contrary to previous conclusions (Nivre & Nilsson 2004), results indicate that subtrees may be a more suitable solution for some MWE types, specially when looking at MWE-aware parsing evaluation metrics (Section 7). In this study, the words-with-spaces representation is shown to have a harming effect on the types where it increases lexical sparsity, such as in Turkish light-verb constructions.

Candito & Constant (2014) explore several orchestrations for combining syntactic parsing and continuous MWE identification in French, distinguishing syntactically regular from irregular multiword constructions. In particular, they experimented with an off-the-shelf graph-based parser that was learned from an MWE-aware treebank where the subtrees representing regular and irregular expressions have their usual labels suffixed by the POS of the MWE, as shown in Figure 5. They showed on-par results with different pipeline variants.

Nasr et al. (2015) focus on ambiguous compound grammatical words in French of the form ADV+*que* and *de*+DET. While these represent a limited scope, such constructions are pervasive and hard to identify without access to syntactic information, because its component words can co-occur by chance. For instance, the two sentences below have the same sequences of POS and similar lexical units, but the first one contains an MWE whereas the second one does not:

(5) Je chante *bien que* je sois triste.
I sing well that I am sad
'I sing even though I am sad'

(6) Je pense *bien que* je suis triste.
I think well that I am sad
'Indeed, I think that I am sad.'

In order to deal with these constructions, the training treebank is modified similarly to Candito & Constant (2014), splitting MWEs originally represented as words with spaces into two tokens linked by a special dependency. For example, since *bien que* functions as a conjunction, the conjunction *que* becomes the head, modified by the adverb *bien*. Using a standard graph-based dependency parser, the authors evaluate the identification of the target MWEs on a dedicated dataset. As described in Section 6.3, the use of subcategorization frame information for verbs, coming from an external lexicon, improves the results.

5.3 MWE-aware parsing models

The models discussed up to now have the advantage of being simple and fast to deploy. Provided that the training treebank contains MWEs in a suitable representation (which can be manually or automatically converted), the parsing algorithm itself does not need to be changed to accommodate MWEs. These approaches achieve reasonably good results, specially if compared to MWE systems based on purely sequential models. However, they often use language-specific or treebank-specific workarounds and are not always generalizable. Therefore, some recent contributions focus on designing parsing models that are truly aware of MWEs in the model, with promising results.

In the framework of constituency parsing, Green et al. (2011) propose and evaluate an MWE-aware parser based on tree substitution grammars (TSGs). This work was latter extended, comparing the TSG with a PCFG model enriched with a factorized lexicon (Green et al. 2013). The authors apply these models to MWE-rich treebanks for French and Arabic, showing gains for both parsing and MWE identification. The authors state that TSGs are more powerful than PCFGs, being able to store lexicalized tree fragments. They are therefore more suitable for idiomatic MWEs, whose particular syntactic analysis requires larger contexts to be predicted.

Along the same lines, Le Roux et al. (2014) design a joint parsing and MWE identification model based on dual decomposition. In this work, however, a specialized sequence model performs lexical segmentation of MWEs. The MWE identification module uses conditional random fields, while the parsing module uses a PCFG-LA also including MWE identification, using the approach of Green et al. (2013). Both models are combined using penalty vectors that are updated in an iterative way. In other words, until reaching consensus on MWE identification, the MWE identifier and parser analyse the input sentence. If the systems do not agree, they are penalized in proportion to the difference between the given solution and the average solution. This model reaches impressive performance on the French treebank, reaching an MWE identification F-score of up to 82.4% on the test set.

Constant & Nivre (2016) propose a new dependency parsing system that jointly performs syntactic analysis and lexical segmentation (including MWE identification). The authors design and evaluate a transition-based parser using two synchronized stacks: one for syntactic parsing and another for lexical segmentation. The synchronization of both stacks is guaranteed by a unique Push transition which pushes the first element of the buffer on both stacks. The parser models MWE-dedicated transitions Merge$_N$ and Merge$_F$, which respectively create new merged lexical nodes for regular MWEs and lexico-syntactic nodes for fixed MWEs. An additional Complete transition marks that a given lexical node has been fully parsed (while being potentially implicit). This approach obtains results that compare with or exceed state-of-the-art performance on French and English MWE-rich treebanks. Finally, the authors show that lexical information can guide parsing, leading to slightly better syntactic trees. The converse assumption does not seem to hold, though, as adding syntactic information to a purely lexical parser tends to slightly degrade its performance.

6 Integration of lexical resources

Lexical resources are large-scale repositories of information typically about simple words, more rarely about MWEs. They can play different roles with respect to statistical MWE-aware parsing, and in this section we discuss three of them. We show how lexical information can help in general to resolve parsing ambiguities (Section 6.1). Then, we focus on the availability of lexical information within pipeline approaches (Section 6.2). Finally, we shift the emphasis to the effect of lexical resources on MWE identification rather than on parsing itself (Section 6.3).

6.1 General integration of lexical resources in statistical parsers

Statistical parsers have several drawbacks due to the limited size of available gold standard treebanks used for training. Many words in the datasets are infrequent, which makes it very difficult to learn relevant (lexical) regularities. In addition, when parsing an unseen text, some words are simply absent from the training dataset, which negatively impacts parsing accuracy. Experiments with different solutions have been undertaken within the parsing community, notably by incorporating external resources mostly (but not only) learned automatically from large raw corpora.

The use of word clusters is one method to deal with the lexical sparsity issue. Clusters (e.g., Brown clusters), consist of groups of words occurring in the same context. Replacing words by clusters or using clusters as features has each been shown to improve parsing accuracy (Koo et al. 2008; Candito & Seddah 2010). Pairs of words that co-occur frequently in large corpora tend to be related syntactically. The provision of information about such lexical affinities to the parser has been shown to usefully support syntactic attachment decisions. Lexical affinities might be integrated using either soft constraints (Bansal & Klein 2011; Mirroshandel et al. 2012) or hard ones (Mirroshandel & Nasr 2016). The deep learning revolution has opened new perspectives to help handle lexical sparsity, as words are represented as continuous space vectors (i.e., word embeddings) learned from large corpora. Words having similar syntactic behaviors have vectors that are geometrically close to each other (Durrett & Klein 2015; Dyer et al. 2015).

The use of external lexicons has also turned out to be of great interest, notably for dependency parsing. For instance, Candito et al. (2010) successfully use the MElt tagger (Denis & Sagot 2012), thereby incorporating features based on a large-scale morphological lexicon. The integration of hard constraints based on syntactic lexicons was also shown to have a positive impact (Mirroshandel et al. 2013).

6.2 MWE resources help parsing

We now give examples of MWE lexical resource integration using a pipeline approach (Section 4) in which MWEs are replaced by their syntactic heads. We do so on two levels: general NP chunking and technical terms.

On the chunking level, replacing chunks with their head words reduces parsing complexity considerably. According to experiments carried out by Prins (2005), parsing performance also increases slightly. However, experiments on technical terms have not confirmed this hypothesis. In other words, replacing chunks with their heads does not necessarily lead to improved results in other settings.

Weeds et al. (2007) used a substitution approach (Section 4.1.2) for term identification in the domain of biomedical research, where gene and protein names in particular are often MWEs. Because taggers, unless they are trained on the domain, perform very poorly, they report better results when replacing technical terms with their head, using a large lexicon of domain terms.[5]

A comparable example is the situation in which a sentence such as *… he did not see the traffic_N light_V* is POS-tagged incorrectly (*light_V* instead of *light_N*). Here, a pipeline substitution approach relying on an MWE lexicon can clearly improve results. This improvement is passed on to the subsequent parsing step. When domain-adapted taggers are available, though, the advantages of the substitution approach tend to disappear. The performance of adapted taggers is often comparable or slightly higher than that of the substitution approach, as tagging accuracy of technical terms increases. In short, sometimes it is better to adapt statistical models (in this case, a domain-adapted tagger) rather than using lexical resources (in this case, an MWE gazetteer of the domain).

Schneider (2014) conducted an experiment using LT-TTT2, an off-the-shelf rule-based named entity recognizer (Grover 2008) on the standard evaluation suite GREVAL (Carroll et al. 2003) with the same approach of replacing multiword named entities by the head of the MWE. The performance of the substitution approach was slightly worse than when leaving the MWE unchanged. Also this experiment did confirm that statistically motivated resources are usually better than purely lexical resources.

6.3 Lexical resources help MWE-aware parsing

Having discussed the effect of lexical resources on parsing accuracy, we now turn to two different ways to use them as a source of features for dependency parsers, to help MWE identification as well as parsing accuracy.

[5] Such a lexicon is often referred to as "gazetteer".

The first is to use MWE lexicons to alleviate the low coverage of MWEs in the training dataset. The idea is to perform an MWE pre-segmentation of the input text by lexicon lookup. The pre-segmentation, encoded in an IOB-like format, is then used as source of features during MWE-aware parsing, either in the parser itself for joint approaches (Candito & Constant 2014), or in the MWE tagger applied before parsing in pipeline approaches (Constant et al. 2012; Constant, Candito, et al. 2013).

One advantage of using soft constraints like features is their ability to handle ambiguous MWEs. Let us take the sequence *up to*, which can be either a complex preposition (*no more than*) or an accidental co-occurrence (*look up to the sky*). A naive segmentation will systematically consider it to be an MWE, independently of the context. However, a better decision can be made taking the context (i.e., the set of other features) into account. Using a joint approach on the French treebank, Candito & Constant (2014) managed to gain around 4 points in terms of tagged MWE identification F-score using such lexicon-based features: F1 = 74.5 (with) vs. F1 = 70.7 (without). We should recall, however, that their approach is limited to continuous MWEs.

A second method proposed by Nasr et al. (2015) is to incorporate subcategorization frame information, derived from a syntactic lexicon, as features in a joint parser. This was used to improve the resolution of ambiguities between grammatical compound MWEs and accidental co-occurrences. An example is the French sequence *bien que* which is either a multiword conjunction ('although') or an adverb ('well') followed by a relative conjunction ('that'), as exemplified in Section 5.2. This ambiguity may be resolved using information about the verb in the syntactic neighborhood. The authors included specific features indicating whether a given verb accepts a given complement: *manger* ('to eat') −QUE −DE, *penser* ('to think') +QUE −DE, *boire* ('to drink') −QUE −DE, *parler* ('to speak') −QUE +DE. In particular, they show for French that there is a 1-point gain in F-score, 85.24 (without) vs. 86.41 (with), for MWEs of the form ADV+*que* (ADV+*that*). The effect is spectacular for compounds of the form *de*+DET, that display a 15-point gain: 75.00 (without) vs. 84.67 (with).

7 Evaluation

Evaluating a syntactic parser generally consists in comparing the output to reference (gold-standard) parses from a manually labeled treebank. In the case of constituency parsing, a constituent is treated as correct if there exists a constituent in the gold standard parse with the same labels, starting and ending points. These parsers are traditionally evaluated through precision, recall and F-score (Black et al. 1991; Sekine & Collins 1997).

In standard dependency parsing with single-head constraint[6], the number of dependencies produced by a parser is equal to the number of total dependencies in the gold-standard parse tree. Common metrics to evaluate these parsers include the percentage of tokens with correct head, called UNLABELLED ATTACHMENT SCORE (UAS), and the percentage of tokens with correct head *and* dependency label, called LABELED ATTACHMENT SCORE (LAS) (Buchholz & Marsi 2006; Nilsson et al. 2007).

The evaluation of MWE-aware parsers and the evaluation of whether or not MWE pre-identification helps improving the parsing quality should be carefully carried out. As stated in previous sections, in most works where MWE identification is realized before parsing, the MWEs are merged into single tokens. As a result, the common metrics for parsing evaluation given above become problematic for measuring the impact of MWE identification on parsing performance (Eryiğit et al. 2011). For example, in dependency parsing, the concatenation of MWEs into single units decrements the total number of evaluated dependencies. It is thus possible to obtain different scores without actually changing the quality of the parser, but simply the representation of the results. Instead of UAS and LAS metrics, the attachment scores on the surrounding structures, namely UAS_{surr} and LAS_{surr} (i.e., the accuracy on the dependency relations excluding the ones between MWE elements) are more appropriate for extrinsic evaluation of the impact of MWE identification on parsing. Similar considerations apply to constituency parsing.

Figure 9 provides two example sequences for the phenomena discussed above; one containing a continuous MWE (on the left side) and another one containing a non-continuous MWE (on the right side). The dependency trees in this figure provide the gold standard unlabeled dependency relations for both examples. Correctly predicted dependencies are presented with check marks (✓) over the relations, whereas the wrongly predicted dependencies are presented with a cross mark (✗). The continuous MWE of the left side sequence consists of three tokens (w_4, w_5 and w_6). In other words, the two dependency relations of the overall sequence belong to the relations between MWE elements. The non-continuous MWE of the right side sequence consists of two tokens (w_3 and w_6).

The first examples of each column (A and E) show the success of a dependency parser without any prior MWE identification process. In the remaining settings, an MWE identifier is run over the given sequence before parsing. Both

[6] Each dependent node has at most one head in the produced dependency tree.

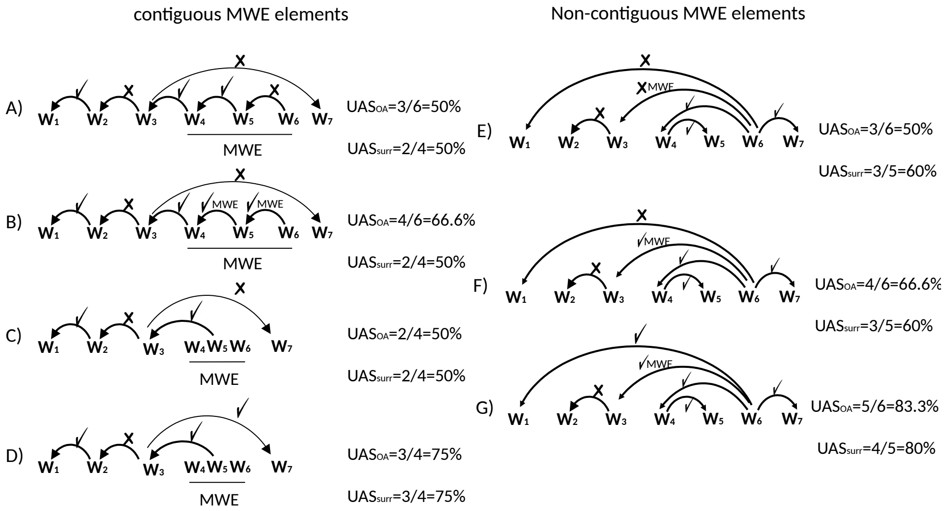

Figure 9: Extrinsic evaluation examples of the impact of MWE identification on dependency parsing performances

the overall unlabeled accuracy UAS_{OA} and the accuracy of the surrounding structures UAS_{surr} are provided next to the trees. Examples (B), (C) and (D) show the correctly detected relations by applying an MWE identifier prior to the syntactic parsing. In (C) and (D), the detected MWE is combined into a single unit ($w_4w_5w_6$) whereas in (B), the detected MWE is represented as a subtree.

In (A), (B) and (C), although the parser success does not change on detecting the syntactic dependencies, UAS_{OA} is affected by the total number of evaluated dependencies, whereas UAS_{surr} remains stable, as expected. In (D), MWE identification helps the parser to detect one more dependency relation, which is reflected in UAS_{surr}. Similarly, in (F), the pre-identification of "w_3 - w_6" MWE has no impact on the parser's performance. Although this can be directly observed by UAS_{surr} (60%), UAS_{OA} mistakenly gives the impression of an improvement in parsing performance (50% $\Longrightarrow$ 66.6%). This is because in this setting (second column of Figure 9) UAS_{OA} evaluates the performance of MWE pre-identification and dependency parsing as a whole. In (G), the parser performs better after MWE identification, which is again reflected in the surrounding structure evaluation.

Although UAS_{surr} and LAS_{surr} are valuable scores for measuring the impact of identifying different MWE types on parsing performance, they are troublesome with automatic MWE identification, when gold-standard MWE segmentation is not available. Then, erroneous MWE identification would degrade parsing scores on the surrounding dependencies. An alternative solution is to detach the concatenated MWE components (if any) into a dependency or constituency subtree (Candito & Constant 2014; Eryiğit et al. 2011). This way, the standard evaluation scores UAS and LAS are still applicable in all different orchestration scenarios, for both continuous and non-continuous MWEs, successfully assessing the performance of joint syntactic parsing and MWE identification as a whole.

8 Conclusions

In this chapter, we elaborated upon several approaches for combining MWE processing with statistical parsing to yield statistical MWE-aware parsing. These approaches depend on different parameters such as MWE representation, orchestration and external resource integration. First of all, the selected MWE representation combined with syntactic analysis have a strong impact on the system implementation, since the more elaborated and hence more linguistically expressive the representation is, the more complex the computational system has to be. Representations vary from simple words with spaces to multilayer structures. The timing of MWE identification with respect to syntactic parsing, namely orchestration, is a crucial feature that needs to be carefully taken into account when designing a statistical MWE-aware parser, as the best choice partly depends on MWE type under consideration. MWE identification may be performed before, after, or during parsing. The first two were discussed under the rubric "pipeline" approaches in Section 4; the third, under "joint" approaches, in Section 5. Last, we showed that the use of external resources is another important feature that is required to handle the sparsity problem, not only to support syntactic attachment decisions, but also MWE identification.

Although it is difficult to draw hard and fast conclusions, it seems that further investigation of dedicated MWE-aware parsing models is called for. Such models can benefit from joint modeling of closely related tasks, with information from one layer helping to disambiguate the other. Joint approaches seem to offer a very promising line of research, as has been shown for other NLP tasks: e.g., joint POS tagging and parsing (Bohnet et al. 2013), joint syntactic and semantic parsing (Henderson et al. 2013). Such approaches are now becoming prominent in NLP alongside the deep learning revolution. In fact, most joint approaches to statistical MWE-aware parsing are not truly joint, as they consist of workaround solutions. We saw how many studies investigated the use of off-the-shelf parsers by modifying training data, thus making the datasets MWE-aware. Truly joint systems are rarer, requiring the use of specific grammatical formalisms for constituency parsing or the development of new dependency parsing mechanisms dedicated to MWE identification.

As a consequence, there is much ground for future work. However, special emphasis should be given to the development of MWE-rich treebanks. Not only are these resources lacking for many languages, but also the representation and covered MWE types vary considerably among different resources. We believe that the development of new MWE-aware parsing models and resources would enable satisfactory solutions for this hard problem. Such solutions could then be further integrated into downstream applications, taking a significant step towards semantic processing of MWEs, and thus of a key element of language itself.

Acknowledgements

This work was supported by the PARSEME action (COST IC1207). It was also partially funded by the French National Research Agency (ANR) through the PARSEME-FR project (ANR-14-CERA-0001) and by a TUBITAK 1001 grant (no: 112E276).

References

Angelov, Krasimir. 2019. Multiword expressions in multilingual applications within the Grammatical Framework. In Yannick Parmentier & Jakub Waszczuk (eds.), *Representation and parsing of multiword expressions: Current trends*, 127–146. Berlin: Language Science Press. DOI:10.5281/zenodo.2579041

Arun, Abhishek & Frank Keller. 2005. Lexicalization in crosslinguistic probabilistic parsing: The case of French. In *Proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL'05)*, 306–313. Ann Arbor, Michigan: Association for Computational Linguistics. http://aclweb.org/anthology/P05-1038.

Bansal, Mohit & Dan Klein. 2011. Web-scale features for full-scale parsing. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*, 693–702. Portland, Oregon. http://www.aclweb.org/anthology/P11-1070.

Bejček, Eduard, Jarmila Panevová, Jan Popelka, Pavel Straňák, Magda Ševčíková, Jan Štěpánek & Zdeněk Žabokrtskỳ. 2012. Prague Dependency Treebank 2.5 – A revisited version of PDT 2.0. In *Proc. of COLING 2012*, 231–246. Bombay, India.

Black, E., S. Abney, S. Flickenger, C. Gdaniec, C. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini & T. Strzalkowski. 1991. Procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the workshop on Speech and Natural Language*, 306–311. Pacific Grove, California: Association for Computational Linguistics. DOI:10.3115/112405.112467

Blunsom, Phil & Timothy Baldwin. 2006. Multilingual deep lexical acquisition for HPSGs via supertagging. In *Proceedings of the 2006 conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, 164–171. Sydney.

Bohnet, Bernd, Joakim Nivre, Igor Boguslavsky, Richard Farkas, Filip Ginter & Jan Hajic. 2013. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics (TACL)* 1. 415–428.

Buchholz, Sabine & Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on Computational Natural Language Learning (CoNLL-X 2006)*, 149–164. New York, NY.

Cafferkey, Conor, Deirdre Hogan & Josef van Genabith. 2007. Multi-word units in treebank-based probabilistic parsing and generation. In *Proceedings of the international conference on Recent Advances in Natural Language Processing (RANLP 2007)*. Borovets, Bulgaria.

Candito, Marie & Mathieu 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)*, 743–753. Baltimore, Maryland: Association for Computational Linguistics. http://aclweb.org/anthology/P14-1070.

Candito, Marie & Benoît Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *Proc. of the 11th International Conference on Parsing Technologies (IWPT'09)*, 138–141. Paris, France: Association for Computational Linguistics. http://www.aclweb.org/anthology/W09-3821.

Candito, Marie, Joakim Nivre, Pascal Denis & Enrique Henestroza Anguiano. 2010. Benchmarking of statistical dependency parsers for French. In *Proceedings of COLING 2010, 23rd international conference on Computational Linguistics , posters volume*, 108–116. Beijing, China.

Candito, Marie & Djamé Seddah. 2010. Parsing word clusters. In *Proceedings of the NAACL HLT 2010 first workshop on Statistical Parsing of Morphologically-Rich Languages*, 76–84. Los Angeles, California.

Carroll, John, Guido Minnen & Edward Briscoe. 2003. Parser evaluation:Using a grammatical relation annotation scheme. In Anne Abeillé (ed.), *Treebanks: Building and using parsed corpora*, 299–316. Dordrecht: Kluwer.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, 132–139. Seattle, Washington.

Collins, Michael. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th annual meeting of the Association for Computational Linguistics (ACL 1996)*, 184–191. Santa Cruz, California.

Collins, Michael. 1999. *Head-driven statistical models for natural language parsing*. Philadelphia, PA: University of Pennsylvania Ph.D. thesis.

Constant, Mathieu, Marie Candito & Djamé Seddah. 2013. The LIGM-Alpage architecture for the SPMRL 2013 shared task: Multiword expression analysis and dependency parsing. In *Proceedings of the fourth workshop on Statistical Parsing of Morphologically-Rich Languages*, 46–52. Seattle, Washington, USA: Association for Computational Linguistics. http://aclweb.org/anthology/W13-4905.

Constant, Mathieu, Joseph Le Roux & Anthony Sigogne. 2013. Combining compound recognition and PCFG-LA parsing with word lattices and conditional random fields. *ACM Transaction on Speech and Language Processing (TSLP), Special Issue on MWEs* 10(3).

Constant, Mathieu, Joseph Le Roux & Nadi Tomeh. 2016. Deep lexical segmentation and syntactic parsing in the easy-first dependency framework. In *Proceedings of the 15th annual conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, 1095–1101. San Diego, California.

Constant, Mathieu & 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*, vol. 1: *Long papers*, 161–171. Berlin, Germany: Association for Computational Linguistics. http://www.aclweb.org/anthology/P16-1016.

Constant, Mathieu, Anthony Sigogne & 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*, vol. 1 (ACL '12), 204–212. Jeju Island, Korea: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=2390524.2390554.

Denis, Pascal & Benoît Sagot. 2012. Coupling an annotated corpus and a lexicon for state-of-the-art POS tagging. *Language Resources and Evaluation* 46(4). 721–736. DOI:10.1007/s10579-012-9193-0

Durrett, Greg & Dan Klein. 2015. Neural CRF parsing. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th*

International Joint Conference on Natural Language Processing, vol. 1: Long Papers, 302–312. Beijing.

Dyer, Chris, Miguel Ballesteros, Wang Ling, Austin Matthews & Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proc. of ACL 2015*, 334–343. Beijing.

Dyvik, Helge, Paul Meurer, Victoria Rosén, Koenraad De Smedt, Petter Haugereid, Gyri Smørdal Losnegaard, Gunn Inger Lyse & Martha Thunes. 2016. NorGramBank: A ‘deep’ treebank for Norwegian. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the tenth international conference on Language Resources and Evaluation (LREC 2016)*, 3555–3562. Portorož, Slovenia. http://www.lrec-conf.org/proceedings/lrec2016/summaries/943.html.

Eisner, Jason M. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational Linguistics (ACL 1996)*, 340–345. Santa Cruz, California.

Eryiğit, Gülşen, Tugay İlbay & Ozan Arkan Can. 2011. Multiword expressions in statistical dependency parsing. In *Proceedings of the second workshop on Statistical Parsing of Morphologically Rich Languages*, 45–55. Dublin, Ireland.

Fazly, Afsaneh, Paul Cook & Suzanne Stevenson. 2009. Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics* 35(1). 61–103.

Foufi, Vasiliki, Luka Nerima & Eric Wehrli. 2019. Multilingual parsing and MWE detection. In Yannick Parmentier & Jakub Waszczuk (eds.), *Representation and parsing of multiword expressions: Current trends*, 217–237. Berlin: Language Science Press. DOI:10.5281/zenodo.2579047

Głowińska, Katarzyna & Adam Przepiórkowski. 2010. The design of syntactic annotation levels in the National Corpus of Polish. In *Proc. of the seventh international conference on Language Resources and Evaluation (LREC 2010).* Valletta, Malta.

Green, Spence, Marie-Catherine de Marneffe, John Bauer & Christopher D. Manning. 2011. Multiword expression identification with tree substitution grammars: A parsing tour de force with French. In *Proc. of EMNLP 2011*, 725–735. Edinburgh.

Green, Spence, Marie-Catherine de Marneffe & Christopher D. Manning. 2013. Parsing models for identifying multiword expressions. *Computational Linguistics* 39(1). 195–227.

Gross, Maurice. 1984. Lexicon-grammar and the syntactic analysis of French. In *Proceedings of the 10th international conference on Computational Linguistics and 22nd annual meeting of the Association for Computational Linguistics*, 275–282. Stanford, California, USA: Association for Computational Linguistics.

Grover, Claire. 2008. *LT-TTT2: Example pipelines documentation.* Tech. rep. Edinburgh Language Technology Group.

Henderson, James, Paola Merlo, Ivan Titov & Gabriele Musillo. 2013. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational Linguistics* 39(4). 949–998.

Kato, Akihiko, Hiroyuki Shindo & Yuji Matsumoto. 2016. Construction of an English dependency corpus incorporating compound function words. In *Proceedings of the tenth international conference on Language Resources and Evaluation (LREC 2016).* Portorož, Slovenia: European Language Resources Association (ELRA).

Koo, Terry, Xavier Carreras & Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, 595–603. Columbus, Ohio.

Korkontzelos, Ioannis & Suresh Manandhar. 2010. Can recognising multiword expressions improve shallow parsing? In *Proc. of the 11th annual conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies NAACL/HLT 2010*, 636–644. Los Angeles, California.

Le Roux, Joseph, Antoine Rozenknop & Mathieu Constant. 2014. Syntactic parsing and compound recognition via dual decomposition: Application to French. In *Proc. of COLING 2014.* Dublin, Ireland.

Lehmann, Hans Martin & Gerold Schneider. 2011. A large-scale investigation of verb-attached prepositional phrases. In S. Hoffmann, P. Rayson & G. Leech (eds.), *Studies in variation, contacts and change in English, volume 6: Methodological and historical dimensions of corpus linguistics.* Helsinki: Varieng.

McDonald, Ryan, Kevin Lerman & Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the tenth conference on Computational Natural Language Learning (CoNLL-X)*, 216–220. New York City: Association for Computational Linguistics. http://www.aclweb.org/anthology/W/W06/W06-2932.

Miller, George Armitage. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review* 63. 81–97.

Mirroshandel, Seyed Abolghasem & Alexis Nasr. 2016. Integrating selectional constraints and subcategorization frames in a dependency parser. *Computional Linguistics* 42(1). 55–90.

Mirroshandel, Seyed Abolghasem, Alexis Nasr & Joseph Le Roux. 2012. Semi-supervised dependency parsing using lexical affinities. In *Proceedings of the 50th annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, 777–785. Jeju Island, Korea: Association for Computational Linguistics. http://aclweb.org/anthology/P12-1082.

Mirroshandel, Seyed Abolghasem, Alexis Nasr & Benoît Sagot. 2013. Enforcing subcategorization constraints in a parser using sub-parses recombining. In *Proceedings of the 2013 conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies*, 239–247. Atlanta, Georgia: Association for Computational Linguistics.

Nagy T., István & Veronika Vincze. 2014. VPCTagger: Detecting verb-particle constructions with syntax-based methods. In *Proceedings of the EACL 2014 workshop on MWEs*, 17–25. Gothenburg.

Nakagawa, Tetsuji. 2007. Multilingual dependency parsing using global features. In *Proceedings of the CoNLL shared task session of EMNLP-CoNLL 2007*, 952–956. Prague, Czech Republic: Association for Computational Linguistics. http://www.aclweb.org/anthology/D07-1100.

Nasr, Alexis, Frederic Bechet, Jean-Francois Rey, Benoit Favre & Joseph Le Roux. 2011. MACAON: An NLP tool suite for processing word lattices. In *Proceedings of ACL 2011 demonstrations*. Portland, Oregon.

Nasr, Alexis, Carlos Ramisch, José Deulofeu & André Valli. 2015. Joint dependency parsing and multiword expression tokenisation. In *53rd annual meeting of the Association for Computational Linguistics*, 1116–1126. Beijing, China.

Nilsson, Jens, Sebastian Riedel & Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of EMNLP/CoNLL 2007 CoNLL shared tasks session*, 915–932.

Nivre, Joakim. 2014. *Transition-based parsing with multiword expressions*. Athens.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty & Dan Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th international conference on Language Resources and Evaluation (LREC 2016)*. Portorož, Slovenia.

Nivre, Joakim, Johan Hall & Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of CoNLL 2004*, 49–56. Boston, Massachusetts.

Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov & Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13(02). 95–135.

Nivre, Joakim & Jens Nilsson. 2004. Multiword units in syntactic parsing. *Proceedings of Methodologies and Evaluation of Multiword Units in Real-World Applications (MEMURA).*

Pawley, Andrew & Frances Hodgetts Syder. 1983. Two puzzles for linguistic theory : Native-like selection and native-like fluency. In J. C. Richards & R. W. Schmidt (eds.), *Language and communication*, 191–226. London: Longman.

Prins, Robbert. 2005. *Finite-state pre-processing for natural language analysis.* Behavioral & Cognitive Neurosciences (BCN) research school, University of Groningen dissertation.

Ramshaw, Lance A. & Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the 3rd ACL workshop on Very Large Corpora*, 82–94.

Ronan, Patricia & Gerold Schneider. 2015. Determining light verb constructions in contemporary British and Irish English. *International Journal of Corpus Linguistics* 20(3). 326–354.

Rosén, Victoria, Gyri Smørdal Losnegaard, Koenraad De Smedt, Eduard Bejček, Agata Savary, Adam Przepiórkowski, Petya Osenova & Verginica Barbu Mititelu. 2015. A survey of multiword expressions in treebanks. In *Proc. of 14th international workshop on Treebanks and Linguistic Theories (TLT 2015)*, 179–193. Warsaw, Poland.

Sag, Ivan, Timothy Baldwin, Francis Bond, Ann Copestake & Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the 3rd international conference on Computational Linguistics and Intelligent Text Processing* (Lecture Notes in Computer Science 2276), 1–15. Springer.

Sagot, Benoît & Pierre Boullier. 2005. From raw corpus to word lattices: Robust pre-parsing processing with SxPipe. *Archives of Control Sciences* 15(4). 653–662.

Schneider, Gerold. 2008. *Hybrid long-distance functional dependency parsing*. Institute of Computational Linguistics, University of Zurich Doctoral Thesis.

Schneider, Gerold. 2012. Using semantic resources to improve a syntactic dependency parser. In *Proceedings of Semantic Relations II workshop (SEM-II) at LREC 2012*, 67–76. Istanbul, Turkey.

Schneider, Gerold. 2014. Improving PP attachment in a hybrid dependency parser using semantic, distributional, and lexical resources. In *Second PARSEME meeting*. Athens, Greece.

Schneider, Nathan, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T. Mordowanec, Henrietta Conrad & Noah A. Smith. 2014. Comprehensive annotation of multiword expressions in a social web corpus. In *Proceedings of* *the ninth international conference on Language Resources and Evaluation (LREC 2014)*, 456–461. Reykyavik.

Seddah, Djamé, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska & Eric Villemonte de la Clérgerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the fourth international workshop on Statistical Parsing of Morphologically-Rich Languages (SPRML IV)*. Seattle, WA.

Sekine, Satoshi & Michael Collins. 1997. *EVALB bracket scoring program*. http://www.%20cs.%20nyu.%20edu/cs/projects/proteus/evalb.

Seretan, Violeta. 2011. *Syntax-based collocation extraction* (Text, Speech and Language Technology 44). Dordrecht: Springer.

Sinclair, John. 1991. *Corpus, concordance, collocation*. Oxford: Oxford University Press.

Tesnière, Lucien. 1959. *Eléments de syntaxe structurale*. Klincksieck.

Tjong Kim Sang, Erik F. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th conference on Natural Language Learning (CoNLL) – volume 20*, 1–4. Taipei, Taiwan.

Tomasello, Michael. 1998. Cognitive linguistics. In W. Bechtel & G. Graham (eds.), *A companion to cognitive science*. Basil Blackwell.

Tu, Yuancheng & Dan Roth. 2011. Learning English light verb constructions: Contextual or statistical. In *Proceedings of the ACL 2011 workshop on MWEs*, 31–39. Portland, OR.

Vincze, Veronika, István Nagy T. & Gábor Berend. 2011. Multiword expressions and named entities in the Wiki50 corpus. In *Proceedings of the international conference Recent Advances in Natural Language Processing 2011*, 289–295. Hissar, Bulgaria: Association for Computational Linguistics.

Vincze, Veronika, János Zsibrita & István Nagy T. 2013. Dependency parsing for identifying Hungarian light verb constructions. In *Proceedings of the sixth International Joint Conference on Natural Language Processing (IJCNLP)*, 207–215. Nagoya, Japan: Asian Federation of Natural Language Processing. http://aclweb.org/anthology/I13-1024.

Weeds, Julie, James Dowdall, Gerold Schneider, Bill Keller & David Weir. 2007. Using distributional similarity to organise biomedical terminology. In Fidelia

Ibekwe-SanJuan, Anne Condamines & M. Teresa Cabré Castellví (eds.), *Application-driven terminology engineering*. Amsterdam/Philadelphia: Benjamins.

Wray, Alison. 2008. *Formulaic language: Pushing the boundaries*. Oxford University Press.

Yamada, Hiroyasu & Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the International Workshop on Parsing Technologies (IWPT)*, vol. 3, 195–206. Nancy, France.