Strategies for Contiguous Multiword Expression Analysis and Dependency Parsing

Marie Candito
Alpage
Paris Diderot Univ
INRIA
marie.candito@
linguist.univ-paris-diderot.fr

Matthieu Constant
Université Paris-Est
LIGM
CNRS
Matthieu.Constant@
u-pem.fr

Abstract

In this paper, we investigate various strategies to predict both syntactic dependency parsing and contiguous multiword expression (MWE) recognition, testing them on the dependency version of French Treebank (Abeillé and Barrier, 2004), as instantiated in the SPMRL Shared Task (Seddah et al., 2013). Our work focuses on using an alternative representation of syntactically regular MWEs, which captures their syntactic internal structure. We obtain a system with comparable performance to that of previous works on this dataset, but which predicts both syntactic dependencies and the internal structure of MWEs. This can be useful for capturing the various degrees of semantic compositionality of MWEs.

1 Introduction

A real-life parsing system should comprise the recognition of multi-word expressions (MWEs[1]), first because downstream semantic-oriented applications need some marking in order to distinguish between regular semantic composition and the typical semantic non-compositionality of MWEs. Second, MWE information, is intuitively supposed to help parsing.

That intuition is confirmed in a classical but non-realistic setting in which *gold* MWEs are pregrouped (Arun and Keller, 2005; Nivre and Nilsson, 2004; Eryiğit et al., 2011). But the situation is much less clear when switching to automatic MWE prediction. While Cafferkey et al. (2007) report a small improvement on the pure parsing task when using external MWE lexicons to help English parsing, Constant et al. (2012) report results on the joint MWE recognition and parsing task, in which errors in MWE recognition alleviate their positive effect on parsing performance.

While the realistic scenario of syntactic parsing with automatic MWE recognition (either done jointly or in a pipeline) has already been investigated in constituency parsing (Green et al., 2011; Constant et al., 2012; Green et al., 2013), the French dataset of the SPMRL 2013 Shared Task (Seddah et al., 2013) only recently provided the opportunity to evaluate this scenario within the framework of dependency syntax.[2] In such a scenario, a system predicts dependency trees with marked groupings of tokens into MWEs. The trees show syntactic dependencies between semantically sound units (made of one or several tokens), and are thus particularly appealing for downstream semantic-oriented applications, as dependency trees are considered to be closer to predicate-argument structures.

In this paper, we investigate various strategies for predicting from a tokenized sentence both MWEs and syntactic dependencies, using the French dataset of the SPMRL 13 Shared Task. We focus on the use of an alternative representation for those MWEs that exhibit regular internal syntax. The idea is to represent these using regular syntactic internal structure, while keeping the semantic information that they are MWEs.

We devote section 2 to related work. In section 3, we describe the French dataset, how MWEs are originally represented in it, and we present and motivate an alternative representation. Section 4 describes the different architectures we test

[1] Multiword expressions can be roughly defined as continuous or discontinuous sets of tokens, which either do not exhibit full freedom in lexical selection or whose meaning is not fully compositional. We focus in this paper on *contiguous* multiword expressions, also known as "words with spaces".

[2] The main focus of the Shared Task was on predicting both morphological and syntactic analysis for morphologically-rich languages. The French dataset is the only one containing MWEs: the French treebank has the particularity to contain a high ratio of tokens belonging to a MWE (12.7% of non numerical tokens).

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 743–753,
Baltimore, Maryland, USA, June 23-25 2014.

for predicting both syntax and MWEs. Section 5 presents the external resources targeted to improve MWE recognition. We describe experiments and discuss their results in section 6 and conclude in section 7.

2 Related work

We gave in introduction references to previous work on predicting MWEs and constituency parsing. To our knowledge, the first works[3] on *predicting* both MWEs and *dependency trees* are those presented to the SPMRL 2013 Shared Task that provided scores for French (which is the only dataset containing MWEs). Constant et al. (2013) proposed to combine pipeline and joint systems in a reparser (Sagae and Lavie, 2006), and ranked first at the Shared Task. Our contribution with respect to that work is the representation of the internal syntactic structure of MWEs, and use of MWE-specific features for the joint system. The system of Björkelund et al. (2013) ranked second on French, though with close UAS/LAS scores. It is a less language-specific system that reranks n-best dependency parses from 3 parsers, informed with features from predicted constituency trees. It uses no feature nor treatment specific to MWEs as it focuses on the general aim of the Shared Task, namely coping with prediction of morphological and syntactic analysis.

Concerning related work on the representation of MWE internal structure, we can cite the Prague Dependency Bank, which captures both regular syntax of non-compositional MWEs and their MWE status, in two distinct annotation layers (Bejček and Stranak, 2010). Our representation also resembles that of light-verb constructions (LVC) in the hungarian dependency treebank (Vincze et al., 2010): the construction has regular syntax, and a suffix is used on labels to express it is a LVC (Vincze et al., 2013).

3 Data: MWEs in Dependency Trees

The data we use is the SPMRL 13 dataset for French, in dependency format. It contains projective dependency trees that were automatically derived from the latest status of the French Treebank (Abeillé and Barrier, 2004), which consists of constituency trees for sentences from the newspaper *Le Monde*, manually annotated with phrase structures, morphological information, and grammatical functional tags for dependents of verbs. The Shared Task used an enhanced version of the constituency-to-dependency conversion of Candito et al. (2010), with different handling of MWEs. The dataset consists of 18535 sentences, split into 14759, 1235 and 2541 sentences for training, development, and final evaluation respectively.

We describe below the flat representation of MWEs in this dataset, and the modified representation for regular MWEs that we propose.

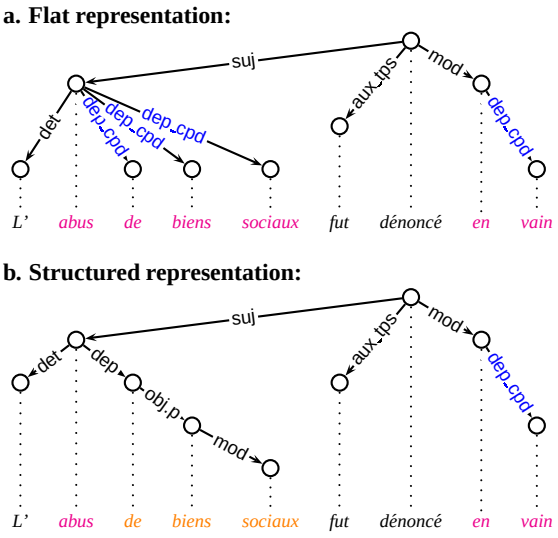

Figure 1: French dependency tree for *L'abus de biens sociaux fut dénoncé en vain* (literally *the misuse of assets social was denounced in vain*, meaning *The misuse of corporate assets was denounced in vain*), containing two MWEs (in red). Top: original flat representation. Bottom: Tree after regular MWEs structuring.

3.1 MWEs in Gold Data: Flat representation

In gold data, the MWEs appear in an expanded flat format: each MWE bears a part-of-speech and consists of a sequence of tokens (hereafter the "components" of the MWE), each having their proper POS, lemma and morphological features. In the dependency trees, there is no "node" for a MWE as a whole, but one node per MWE component (more generally one node per token). The first component of a MWE is taken as the head of the MWE. All subsequent components of the MWE depend on the first one, with the special label `dep_cpd` (hence the name *flat represen-*

[3] Concerning non contiguous MWEs, we can cite the work of Vincze et al. (2013), who experimented joint dependency parsing and light verb construction identification.

tation). Furthermore, the first MWE component bears a feature `mwehead` equal to the POS of the MWE. An example is shown in Figure 1. The MWE *en vain* (*pointlessly*) is an adverb, containing a preposition and an adjective. The latter depends on former, which bears `mwehead=ADV+`.

The algorithm to recover MWEs is: any node having dependents with the `dep_cpd` label forms a MWE with such dependents.

3.2 Alternative representation for regular MWEs

In the alternative representation we propose, irregular MWEs are unchanged and appear as flat MWEs (e.g. *en vain* in Figure 1 has pattern preposition+adjective, which is not considered regular for an adverb, and is thus unchanged). Regular MWEs appear with 'structured' syntax: we modify the tree structure to recover the regular syntactic dependencies. For instance, in the bottom tree of the figure, *biens* is attached to the preposition, and the adjective *sociaux* is attached to *biens*, with regular labels. Structured MWEs cannot be spotted using the tree topology and labels only. Features are added for that purpose: the syntactic head of the structured MWE bears a `regmwehead` for the POS of the MWE (*abus* in Figure 1), and the other components of the MWE bear a `regcomponent` feature (the orange tokens in Figure 1).[4] With this representation, the algorithm to recover regular MWEs is: any node bearing `regmwehead` forms a MWE with the set of direct or indirect dependents bearing a `regcomponent` feature.

3.2.1 Motivations

Our first motivation is to increase the quantity of information conveyed by the dependency trees, by distinguishing syntactic regularity and semantic regularity. Syntactically regular MWEs (hereafter regular MWEs) show various degrees of semantic non-compositionality. For instance, in the French Treebank, *population active* (lit. *active population*, meaning *'working population'*) is a partially compositional MWE. Furthermore, some sequences are both syntactically and semantically regular, but encoded as MWE due to frozen lexical selection. This is the case for *déficit budgétaire* (lit. *budgetary deficit*, meaning *'budget deficit'*), because it is not possible to use *déficit du budget* (*budget deficit*). Our alternative representation distinguishes between syntactic internal regularity and semantic regularity. This renders the syntactic description more uniform and it provides an internal structure for regular MWEs, which is meaningful if the MWE is fully or partially compositional. For instance, it is meaningful to have the adjective *sociaux* attach to *biens* instead of on the first component *abus*. Moreover, such a distinction opens the way to a non-binary classification of MWE status: the various criteria leading to classify a sequence as MWE could be annotated separately and using nominal or scaled categories for each criteria. For instance, *déficit budgétaire* could be marked as fully compositional, but with frozen lexical selection. Further, annotation is often incoherent for the MWEs with both regular syntax and a certain amount of semantic compositionality, the same token sequence (with same meaning) being sometimes annotated as MWE and sometimes not.

More generally, keeping a regular representation would allow to better deal with the interaction between idiomatic status and regular syntax, such as the insertion of modifiers on MWE subparts (e.g. *make a **quick** decision*).

Finally, using regular syntax for MWEs provides a more uniform training set. For instance for a sequence *N1 preposition N2*, though some *external* attachments might vary depending on whether the sequence forms a MWE or not, some may not, and the internal dependency structure $(N1 \rightarrow (preposition \rightarrow N2))$ is quite regular. One objective of the current work is to investigate whether this increased uniformity eases parsing or whether it is mitigated by the additional difficulty of finding the internal structure of a MWE.

	Total nb of MWEs	Nb of regular MWEs *(% of nouns, adverbs, prepositions, verbs)*
train	23658	12569 *(64.7, 19.2, 14.6, 1.5)*
dev	2120	1194 *(66.7, 17.7, 14.7, 0.8)*
test	4049	2051 *(64.5, 19.9, 13.6, 2.0)*

Table 1: Total number of MWEs and number of regular MWEs in training, development and test set (and broken down by POS of MWE).

[4] The syntactic head of a structured MWE may not be the first token, whereas the head token of a flat MWE is always the first one.

3.2.2 Implementation

We developed an ad hoc program for structuring the regular MWEs in gold data. MWEs are first classified as regular or irregular, using regular expressions over the sequence of parts-of-speech within the MWE. To define the regular expressions, we grouped gold MWEs according to the pair [global POS of the MWE + sequence of POS of the MWE components], and designed regular expressions to match the most frequent patterns that looked regular according to our linguistic knowledge. The internal structure for the matching MWEs was built deterministically, using heuristics favoring local attachments.[5] Table 1 shows the proportions of MWEs classified as regular, and thus further structured. About half MWEs are structured, and about two thirds of structured MWEs are nouns.

For predicted parses with structured MWEs, we use an inverse transformation of structured MWEs into flat MWEs, for evaluation against the gold data. When a predicted structured MWE is flattened, all the dependents of any token of the MWE that are not themselves belonging to the MWE are attached to the head component of the MWE.

3.3 Integration of MWE features into labels

In some experiments, we make use of alternative representations, which we refer later as "labeled representation", in which the MWE features are incorporated in the dependency labels, so that MWE composition and/or the POS of the MWE be totally contained in the tree topology and labels, and thus predictable via dependency parsing. Figure shows the labeled representation for the sentence of Figure 1.

For flat MWEs, the only missing information is the MWE part-of-speech: we concatenate it to the `dep_cpd` labels. For instance, the arc from *en* to *vain* is relabeled `dep_cpd_ADV`. For structured MWEs, in order to get full MWE account within the tree structure and labels, we need to incorporate both the MWE POS, and to mark it as

[5]The six regular expressions that we obtained cover nominal, prepositional, adverbial and verbal compounds. We manually evaluated both the regular versus irregular classification and the structuring of regular MWEs on the first 200 MWEs of the development set. 113 of these were classified as regular, and we judged that all of them were actually regular, and were correctly structured. Among the 87 classified as irregular, 7 should have been tagged as regular and structured. For 4 of them, the classification error is due to errors on the (gold) POS of the MWE components.

c. Labeled representation:

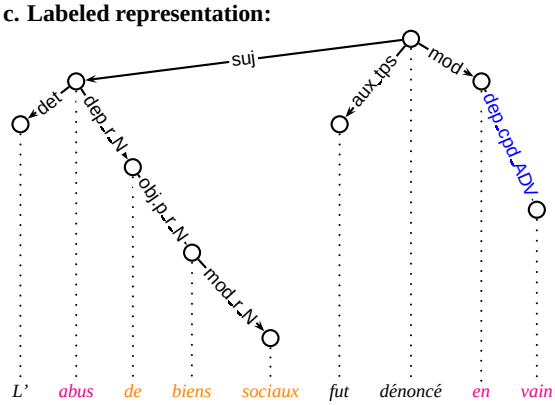

Figure 2: Integration of all MWE information into labels for the example of Figure 1.

belonging to a MWE. The suffixed label has the form `FCT_r_POS`. For instance, in bottom tree of Figure 1, arcs pointing to the non-head components (*de*, *biens*, *sociaux*) are suffixed with `_r` to mark them as belonging to a structured MWE, and with `_N` since the MWE is a noun.

In both cases, this label suffixing is translated back into features for evaluation against gold data.

4 Architectures for MWE Analysis and Parsing

The architectures we investigated vary depending on whether the MWE status of sequences of tokens is predicted via dependency parsing or via an external tool (described in section 5), and this dichotomy applies both to structured MWEs and flat MWEs. More precisely, we consider the following alternative for irregular MWEs:

- IRREG-MERGED: gold irregular MWEs are merged for training; for parsing, irregular MWEs are predicted externally, merged into one token at parsing time, and re-expanded into several tokens for evaluation;

- IRREG-BY-PARSER: the MWE status, flat topology and POS are all predicted via dependency parsing, using representations for training and parsing, with all information for irregular MWEs encoded in topology and labels (as for *in vain* in Figure 2).

For regular MWEs, their internal structure is always predicted by the parser. For instance the unlabeled dependencies for *abus de biens sociaux* are the same, independently of predicting whether it

forms a MWE or not. But we use two kinds of predictions for their MWE status and POS:

- REG-POST-ANNOTATION: the regular MWEs are encoded/predicted as shown for *abus de biens sociaux* in bottom tree of Figure 1, and their MWE status and POS is predicted after parsing, by an external tool.
- REG-BY-PARSER: all regular MWE information (topology, status, POS) is predicted via dependency parsing, using representations with all information for regular MWEs encoded in topology and labels (Figure 2).

Name	prediction of reg MWEs	prediction of irreg MWEs
JOINT	irreg-by-parser	reg-by-parser
JOINT-REG	irreg-merged	reg-by-parser
JOINT-IRREG	irreg-by-parser	reg-post-annot
PIPELINE	irreg-merged	reg-post-annot

Table 2: The four architectures, depending on how regular and irregular MWEs are predicted.

We obtain four architectures, schematized in table 2. We describe more precisely two of them, the other two being easily inferable:

JOINT-REG architecture:

- **training set**: irregular MWEs merged into one token, regular MWEs are structured, and integration of regular MWE information into the labels (`FCT_r_POS`).
- **parsing**: **(i)** MWE analysis with classification of MWEs into regular or irregular, **(ii)** merge of predicted irregular MWEs, **(iii)** tagging and morphological prediction, **(iv)** parsing

JOINT-IRREG architecture:

- **training set**: flat representation of irregular MWEs, with label suffixing (`dep_cpd_POS`), structured representation of regular MWEs without label suffixing.
- **parsing**: **(i)** MWE analysis and classification into regular or irregular, used for MWE-specific features, **(ii)** tagging and morphological prediction, **(iii)** parsing,

We compare these four architectures between them and also with two simpler architectures used by (Constant et al., 2013) within the SPMRL 13 Shared Task, in which regular and irregular MWEs are not distinguished:

Uniform joint architecture: The joint systems perform syntactic parsing and MWE analysis via a single dependency parser, using representations as in 3.3.

Uniform pipeline architecture:

- **training set**: MWEs merged into one token
- **parsing**: **(i)** MWE analysis, **(ii)** merge of predicted MWEs, **(iii)** tagging and morphological prediction, **(iv)** parsing

For each architecture, we apply the appropriate normalization procedures on the predicted parses, in order to evaluate against (i) the pseudo-gold data in structured representation, and (ii) the gold data in flat representation.

5 Use of external MWE resources

In order to better deal with MWE prediction, we use external MWE resources, namely MWE lexicons and an MWE analyzer. Both resources help to predict MWE-specific features (section 5.3) to guide the MWE-aware dependency parser. Moreover, in some of the architectures, the external MWE analyzer is used either to pre-group irregular MWEs (for the architectures using IRREG-MERGED), or to post-annotate regular MWEs.

5.1 MWE lexicons

MWE lexicons are exploited as sources of features for both the dependency parser and the external MWE analyzer. In particular, two large-coverage general-language lexicons are used: the Lefff[6] lexicon (Sagot, 2010), which contains approximately half a million inflected word forms, among which approx. $25,000$ are MWEs; and the DELA[7] (Courtois, 2009; Courtois et al., 1997) lexicon, which contains approx. one million inflected forms, among which about $110,000$ are MWEs. These resources are completed with specific lexicons freely available in the platform Unitex[8]: the toponym dictionary Prolex (Piton et al., 1999) and a dictionary of first names. Note that the lexicons do not include any information on the irregular or the regular status of the MWEs. In order to compare the MWEs present in the lexicons and those encoded in the French treebank, we applied the following procedure (hereafter called lexicon

[6] We use the version available in the POS tagger MElt (Denis and Sagot, 2009).

[7] We use the version in the platform Unitex (http://igm.univ-mlv.fr/~unitex). We had to convert the DELA POS tagset to that of the French Treebank.

[8] http://igm.univ-mlv.fr/~unitex

lookup): in a given sentence, the maximum number of non overlapping MWEs according to the lexicons are systematically marked as such. We obtain about 70% recall and 50% precision with respect to MWE spanning.

5.2 MWE Analyzer

The MWE analyzer is a CRF-based sequential labeler, which, given a tokenized text, jointly performs MWE segmentation and POS tagging (of simple tokens and of MWEs), both tasks mutually helping each other[9]. The MWE analyzer integrates, among others, features computed from the external lexicons described in section 5.1, which greatly improve POS tagging (Denis and Sagot, 2009) and MWE segmentation (Constant and Tellier, 2012). The MWE analyzer also jointly classifies its predicted MWEs as regular or irregular (the distinction being learnt on gold training set, with structured MWEs cf. section 3.2).

5.3 MWE-specific features

We introduce information from the external MWE resources in different ways:

Flat MWE features: MWE information can be integrated as features to be used by the dependency parser. We tested to incorporate the MWE-specific features as defined in the gold flat representation (section 3.1): the *mwehead=POS* feature for the MWE head token, POS being the part-of-speech of the MWE; the *component=y* feature for the non-first MWE component.

Switch: instead or on top of using the `mwehead` feature, we use the POS of the MWE instead of the POS of the first component of a flat MWE. For instance in Figure 1, the token *en* gets `pos=ADV` instead of `pos=P`. The intuition behind this feature is that for an irregular MWE, the POS of the linearly first component, which serves as head, is not always representative of the external distribution of the MWE. For regular MWEs, the usefulness of such a trick is less obvious. The first component of a regular MWE is not necessarily its head (for instance for a nominal MWE with internal pattern adjective+noun), so the switch trick could be detrimental in such cases.[10]

[9]Note that in our experiments, we use this analyzer for MWE analysis only, and discard the POS tagging prediction. Tagging is performed along with lemmatization with the Morfette tool (section 6.1).

[10]We also experimented to use POS of MWE plus suffixes to force disjoint tagsets for single words, irregular MWEs and regular MWEs, but this showed comparable results.

6 Experiments

6.1 Settings and evaluation metrics

MWE Analysis and Tagging: For the MWE analyzer, we used the tool *lgtagger*[11] (version 1.1) with its default set of feature templates, and a 10-fold jackknifing on the training corpus.

Parser: We used the second-order graph-based parser available in *Mate-tools*[12] (Bohnet, 2010). We used the *Anna3.3* version, in projective mode, with default feature sets and parameters proposed in the documentation, augmented or not with MWE-specific features, depending on the experiments.

Morphological prediction: Predicted lemmas, POS and morphology features are computed with Morfette version 0.3.5 (Chrupała et al., 2008; Seddah et al., 2010)[13], using 10 iterations for the tagging perceptron, 3 iterations for the lemmatization perceptron, default beam size for the decoding of the joint prediction, and the Lefff (Sagot, 2010) as external lexicon used for out-of-vocabulary words. We performed a 10-fold jackknifing on the training corpus.

Evaluation metrics: we evaluate our parsing systems by using the standard metrics for dependency parsing: Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS), computed using all tokens including punctuation. To evaluate statistical significance of parsing performance differences, we use eval07.pl[14] with -b option, and then Dan Bikel's comparator.[15] For MWEs, we use the Fmeasure for recognition of untagged MWEs (hereafter FUM) and for recognition of tagged MWEs (hereafter FTM).

6.2 MWE-specific feature prediction

In all our experiments, for the switch trick (section 5.3), the POS of MWE is always predicted using the MWE analyzer. For the flat MWE features, we experimented both with features predicted by the MWE analyzer, and with features predicted using the external lexicons mentioned in section 5.1 (using the lexicon lookup procedure). Both kinds of

[11]http://igm.univ-mlv.fr/~mconstan

[12]http://code.google.com/p/mate-tools/

[13]https://sites.google.com/site/morfetteweb/

[14]http://nextens.uvt.nl/depparse-wiki/SoftwarePage

[15]The compare.pl script, formerly available at www.cis.upenn.edu/ dbikel/

					LABELED REPRES.		STRUCTURED REPRESENTATION			FLAT REPRESENTATION			
ARCHI		MWE feats	swi. irreg	swi. reg	LAS	UAS	LAS	FUM irreg	FTM irreg	LAS	UAS	FUM	FTM
JOINT	bsline	-	-	-	84.5	89.3	87.0	83.6	80.6	84.2	88.1	73.5	70.7
	best	+	+	+	**85.3**	89.7	**87.5**	**85.4**	**82.6**	**85.2**	**88.8**	77.6	74.5
JOINT-IRREG	bsline	-	-	-	84.7	89.4	87.0	83.5	80.3	84.5	88.0	78.3	75.9
	best	+	+	+	85.1	**89.8**	87.4	85.0	81.6	84.9	88.3	**79.0**	**76.5**
JOINT-REG	bsline	-	NA	-	84.2	89.1	86.7	84.2	80.8	84.0	88.0	73.3	70.3
	best	+	NA	+	84.7	89.3	86.9	84.1	80.7	84.6	88.3	76.3	73.2
PIPE LINE	bsline	-	NA	-	84.6	89.2	86.9	84.1	80.7	84.5	87.9	78.8	76.3
	best	-	NA	+	84.7	89.4	87.0	84.2	80.8	84.6	88.1	78.8	76.3

Table 3: Baseline and best results for the four MWE+parsing architectures on the dev set (see text for statistical significance evaluation). The UAS for the structured representation is the same as the one for the labeled representation, and is not repeated.

prediction lead to fairly comparable results, so in all the following, the MWE features, when used, are predicted using the external lexicons.

6.3 Tuning features for each architecture

We ran experiments for all value combinations of the following parameters: (i) the architecture, (ii) whether MWE features are used, whether the switch trick is applied or not (iii) for irregular MWEs and (iv) for regular MWEs.

We performed evaluation of the predicted parses using the three representations described in section 3, namely flat, structured and labeled representations. In the last two cases, the evaluation is performed against an instance of the gold data automatically transformed to match the representation type. Moreover, for the "labeled representation" evaluation, though the MWE information in the predicted parses is obtained in various ways, depending on the architecture, we always map all this information in the dependency labels, to obtain predicted parses matching the "labeled representation". While the evaluation in flat representation is the only one comparable to other works on this dataset, the other two evaluations provide useful information. In the "labeled representation" evaluation, the UAS provides a measure of syntactic attachments for sequences of words, independently of the (regular) MWE status of subsequences. For the sequence *abus de biens sociaux*, suppose that the correct internal structure is predicted, but not the MWE status. The UAS for labeled representation will be maximal, whereas for the flat representation, the last two tokens will count as incorrect for UAS. For LAS, in both cases the three last tokens will count as incorrect if the wrong MWE status is predicted. So to sum up on the "labeled evaluation", we obtain a LAS evaluation for the whole task of parsing plus MWE recognition, but an UAS evaluation that penalizes less errors on MWE status, while keeping a representation that is richer: predicted parses contain not only the syntactic dependencies and MWE information, but also a classification of MWEs into regular and irregular, and the internal syntactic structure of regular MWEs.

The evaluation on "structured representation" can be interpreted as an evaluation of the parsing task plus the recognition of irregular MWEs only: both LAS and UAS are measured independently of errors on regular MWE status (note the UAS is exactly the same than in the "labeled" case).

For each architecture, Table 3 shows the results for two systems: first the baseline system without any MWE features nor switches and immediately below the best settings for the architecture. The JOINT baseline corresponds to a "pure" joint system without external MWE resources (hence the minus sign for the first three columns). For each architecture except the PIPELINE one, differences between the baseline and the best setting are statistically significant ($p < 0.01$). Differences between best PIPELINE and best JOINT-REG are not. Best JOINT has statistically significant difference ($p < 0.01$) over both best JOINT-REG and best PIPELINE. The situation for best JOINT-IRREG with respect to the other three is borderline (with various p-values depending on the metrics).

Concerning the tuning of parameters, it appears that the best setting is to use MWE-features, and switch for both regular and irregular MWEs, except for the pipeline architecture for which results without MWE features are slightly better. So overall, informing the parser with independently pre-

	LABELED REPRESENTATION		STRUCTURED REPRESENTATION				FLAT REPRESENTATION			
SYSTEM	LAS	UAS	LAS	UAS	FUM irreg	FTM irreg	LAS	UAS	FUM	FTM
baseline JOINT	84.13	88.93	86.62	88.93	83.6	79.2	83.97	87.80	73.9	70.5
best JOINT	84.59	89.21	86.92	89.21	85.7	81.4	84.48	88.13	77.0	73.5
best JOINT-IRREG	84.50	89.21	86.97	89.24	86.3	82.1	84.36	87.75	78.6	75.4
best JOINT-REG	84.31	89.0	86.63	89.00	84.5	80.4	84.18	87.95	76.4	73.3
best PIPELINE	84.02	88.83	86.49	88.83	84.4	80.4	83.88	87.33	77.6	74.4

Table 4: Final results on test set for baseline and the best system for each architecture.

dicted POS of MWE has positive impact. The best architectures are JOINT and JOINT-IRREG, with the former slightly better than the latter for parsing metrics, though only some of the differences are significant between the two. It can be noted though, that JOINT-IRREG performs overall better on MWEs (last two columns of table 3), whereas JOINT performs better on irregular MWEs: the latter seems to be beneficial for parsing, but is less efficient to correctly spot the regular MWEs.

Concerning the three distinct representations, evaluating on structured representation (hence without looking at regular MWE status) leads to a rough 2 point performance increase for the LAS and a one point increase for the UAS, with respect to the evaluation against flat representation. This quantifies the additional difficulty of deciding for a regular sequence of tokens whether it forms a MWE or not. The evaluation on the labeled representation provides an evaluation of the full task (parsing, regular/irregular MWE recognition and regular MWEs structuring), with a UAS that is less impacted by errors on regular MWE status, while LAS reflects the full difficulty of the task.[16]

6.4 Results on test set and comparison

We provide the final results on the test set in table 4. We compare the baseline JOINT system with the best system for all four reg/irreg architectures (cf. section 6.3). We observe the same general trend as in the development corpus, but with tinier differences. JOINT and JOINT-IRREG significantly outperform the baseline and the PIPELINE, on labeled representation and flat representation. We can see that there is no significant difference between JOINT and JOINT-IRREG and between JOINT-REG and JOINT-IRREG. JOINT slightly outperforms JOINT-REG ($p < 0.05$). On the structured representation, the two best systems (JOINT and JOINT-IRREG) significantly outperform the other systems ($p < 0.01$ for all; $p < 0.05$ for JOINT-REG).

[16]The slight differences in LAS between the labeled and the flat representations are due to side effects of errors on MWE status: some wrong reattachments performed to obtain flat representation decrease the UAS, but also in some cases the LAS.

	DEV		TEST	
System	UAS	LAS	UAS	LAS
reg/irreg joint	88.79	85.15	88.13	84.48
Bjork13	*88.30*	*84.84*	*87.87*	*84.37*
Const13 pipeline	88.73	85.28	**88.35**	**84.91**
Const13 joint	88.21	84.60	87.76	84.14
uniform joint	**88.81**	**85.42**	87.96	84.59

Table 5: Comparison on dev set of our best architecture with reg/irregular MWE distinction (first row), with the single-parser architectures of (Constant et al., 2013) (Const13) and (Björkelund et al., 2013) (Bjork13). Uniform joint is our reimplementation of Const13 joint, enhanced with mwe-features and switch.

Moreover, we provide in table 5 a comparison of our best architecture with reg/irregular MWE distinction with other architectures that do not make this distinction, namely the two best comparable systems designed for the SPMRL Shared Task (Seddah et al., 2013): the pipeline simple parser based on Mate-tools of Constant et al. (2013) (Const13) and the Mate-tools system (without reranker) of Björkelund et al. (2013) (Bjork13). We also reimplemented and improved the uniform joint architecture of Constant et al. (2013), by adding MWE features and switch. Results can only be compared on the flat representation, because the other systems output poorer linguistic information. We computed statistical significance of differences between our systems and Const13. On dev, the best system is the enhanced uniform joint, but differences are not significant between that and the best reg/irreg joint (1st row) and the Const13 pipeline. But on the test corpus (which is twice bigger), the best system is Const13

	Tasks		LAS	UAS	ALL MWE		REG MWE		IRREG MWE	
System	Parsing	MWE			FUM	FTM	FUM	FTM	FUM	FTM
Our best system (best JOINT)	+	all	85.15	88.78	77.6	74.5	70.8	67.8	85.4	82.6
Uniform pipeline/gold MWEs	+	-	88.73	90.60	-	-	-	-	-	-
CRF-based MWE analyzer	-	all	-	-	78.8	76.3	73.5	71.9	84.2	80.8
JOINT-REG	+	all	84.58	88.34	76.3	73.2	69.3	66.5	84.1	80.7
JOINT-REG/gold irreg. MWE	+	reg.	85.86	89.19	82.9	78.8	70.0	67.2	-	-

Table 6: Comparison with simpler tasks on the flat representation of the development set.

pipeline, with statistically significant differences over our joint systems. So the first observation is that our architectures that distinguish between reg/irreg MWEs do not outperform uniform architectures. But we note that the differences are slight, and the output we obtain is enhanced with regular MWE internal structure. It can thus be noted that the increased syntactic uniformity obtained by our MWE representation is mitigated so far by the additional complexity of the task. The second observation is that currently the best system on this dataset is a pipeline system, as results on test set show (and somehow contrary to results on dev set). The joint systems that integrate MWE information in the labels seem to suffer from increased data sparseness.

6.5 Evaluating the double task with respect to simpler tasks

In this section, we propose to better evaluate the difficulty of combining the tasks of MWE analysis and dependency parsing by comparing our systems with systems performing simpler tasks: i.e. MWE recognition without parsing, and parsing with no or limited MWE recognition, simulated by using gold MWEs. We also provide a finer evaluation of the MWE recognition task, in particular with respect to their regular/irregular status.

We first compare our best system with a parser where all MWEs have been perfectly pre-grouped, in order to quantify the difficulty that MWEs add to the parsing task. We also compare the performance on MWEs of our best system with that achieved by the CRF-based analyzer described in section 5.2. Next, we compare the best JOINT-REG system with the one based on the same architecture but where the irregular MWEs are perfectly pre-identified, in order to quantify the difficulty added by the irregular MWEs. Results are given in table 6. Without any surprise, the task is much easier without considering MWE recognition. We can see that without considering MWE analysis the parsing accuracy is about 2.5 points better in terms of LAS. In the JOINT-REG architecture, assuming gold irregular MWE identification, increases LAS by 1.3 point. In terms of MWE recognition, as compared with the CRF-based analyzer, our best system is around 2 points below. But the situation is quite different when breaking the evaluation by MWE type. Our system is 1 point better than the CRF-based analyzer for irregular MWEs. This shows that considering a larger syntactic context helps recognition of irregular MWEs. The ”weak point” of our system is therefore the identification of regular MWEs.

7 Conclusion

We experimented strategies to predict both MWE analysis and dependency structure, and tested them on the dependency version of French Treebank (Abeillé and Barrier, 2004), as instantiated in the SPMRL Shared Task (Seddah et al., 2013). Our work focused on using an alternative representation of syntactically regular MWEs, which captures their syntactic internal structure. We obtain a system with comparable performance to that of previous works on this dataset, but which predicts both syntactic dependencies and the internal structure of MWEs. This can be useful for capturing the various degrees of semantic compositionality of MWEs. The main weakness of our system comes from the identification of regular MWEs, a property which is highly lexical. Our current use of external lexicons does not seem to suffice, and the use of data-driven external information to better cope with this identification can be envisaged.

References

Anne Abeillé and Nicolas Barrier. 2004. Enriching a french treebank. In *Proceedings of LREC 2004*, Lisbon, Portugal.

Abhishek Arun and Frank Keller. 2005. Lexicalization in crosslinguistic probabilistic parsing: The case of

french. In *Proceedings of ACL 2005*, Ann Arbor, USA.

Eduard Bejček and Pavel Stranak. 2010. Annotation of multiword expressions in the prague dependency treebank. *Language Resources and Evaluation*, 44:7–21.

Anders Björkelund, Özlem Çetinoğlu, Thomas Farkas, Richárdand Müller, and Wolfgang Seeker. 2013. (re)ranking meets morphosyntax: State-of-the-art results from the spmrl 2013 shared task. In *Proceedings of the 4th Workshop on Statistical Parsing of Morphologically Rich Languages: Shared Task*, Seattle, WA.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING 2010*, Beijing, China.

Conor Cafferkey, Deirdre Hogan, and Josef van Genabith. 2007. Multi-word units in treebank-based probabilistic parsing and generation. In *Proceedings of the 10th International Conference on Recent Advances in Natural Language Processing (RANLP'07)*, Borovets, Bulgaria.

Marie Candito, Benoit Crabbé, and Pascal Denis. 2010. Statistical french dependency parsing : Treebank conversion and first results. In *Proceedings of LREC 2010*, Valletta, Malta.

Grzegorz Chrupała, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with morfette. In *Proceedings of LREC 2008*, Marrakech, Morocco. ELDA/ELRA.

Matthieu Constant and Isabelle Tellier. 2012. Evaluating the impact of external lexical resources into a crf-based multiword segmenter and part-of-speech tagger. In *Proceedings of LREC 2012*, Istanbul, Turkey.

Matthieu Constant, Anthony Sigogne, and Patrick Watrin. 2012. Discriminative strategies to integrate multiword expression recognition and parsing. In *Proceedings of ACL 2012*, Stroudsburg, PA, USA.

Matthieu Constant, Marie Candito, and Djamé Seddah. 2013. The ligm-alpage architecture for the spmrl 2013 shared task: Multiword expression analysis and dependency parsing. In *Proceedings of the 4th Workshop on Statistical Parsing of Morphologically Rich Languages: Shared Task*, Seattle, WA.

Blandine Courtois, Mylène Garrigues, Gaston Gross, Maurice Gross, René Jung, Mathieu-Colas Michel, Anne Monceaux, Anne Poncet-Montange, Max Silberztein, and Robert Vivés. 1997. Dictionnaire électronique DELAC : les mots composés binaires. Technical Report 56, University Paris 7, LADL.

Blandine Courtois. 2009. Un système de dictionnaires électroniques pour les mots simples du français. *Langue Française*, 87:11–22.

Pascal Denis and Benoît Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC'09)*, Hong Kong.

Gülşen Eryiğit, Tugay Ilbay, and Ozan Arkan Can. 2011. Multiword expressions in statistical dependency parsing. In *Proceedings of the IWPT Workshop on Statistical Parsing of Morphologically-Rich Languages (SPMRL'11)*, Dublin, Ireland.

Spence Green, Marie-Catherine de Marneffe, John Bauer, and Christofer D. Manning. 2011. Multiword expression identification with tree substitution grammars: A parsing tour de force with french. In *Proceedings of EMNLP 2011*, Edinburgh, Scotland.

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

Joakim Nivre and Jens Nilsson. 2004. Multiword units in syntactic parsing. In *Proceedings of the LREC Workshop : Methodologies and Evaluation of Multiword Units in Real-World Applications (MEMURA)*, Lisbon, Portugal.

Odile Piton, Denis Maurel, and Claude Belleil. 1999. The prolex data base : Toponyms and gentiles for nlp. In *Proceedings of the Third International Workshop on Applications of Natural Language to Data Bases (NLDB'99)*, Klagenfurt, Austria.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In *Proceedings of NAACL/HLT 2006, Companion Volume: Short Papers*, Stroudsburg, PA, USA.

Benoît Sagot. 2010. The lefff, a freely available, accurate and large-coverage lexicon for french. In *Proceedings of LREC 2010*, Valletta, Malta.

Djamé Seddah, Grzegorz Chrupała, Ozlem Cetinoglu, Josef van Genabith, and Marie Candito. 2010. Lemmatization and statistical lexicalized parsing of morphologically-rich languages. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Djamé Seddah, Reut Tsarfaty, Sandra K'ubler, 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, and 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 4th Workshop on Statistical Parsing of Morphologically Rich Languages: Shared Task*, Seattle, WA.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian dependency treebank. In *Proceedings of LREC 2010*, Valletta, Malta.

Veronika Vincze, János Zsibrita, and Istvàn Nagy T. 2013. Dependency parsing for identifying hungarian light verb constructions. In *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP 2013)*, Nagoya, Japan.