Dependency Representations for Lexical Segmentation

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

Joseph Le Roux
LIPN
Université Paris 13 – CNRS
Sorbonne Paris Cité
leroux@lipn.fr

Abstract

This paper investigates the use of dependency structures to represent lexical segmentations including shallow multiword expressions, independently of any syntactic structure. We compare our hierarchical segmenter on several corpora with sequence labelers. Experimental results show comparable scores for flat structures, and open new perspectives for hierarchical representations of deeper constructions, such as nested and interleaved multiword expressions.

1 Introduction

Lexical segmentation is the task of computing, from a sequence of tokens, the corresponding sequence of lexical units. Lexical units are either simple words or multiword expressions. They often correspond to semantic units and their recognition is therefore crucial for many NLP applications like information retrieval or machine translation. For many languages, including most European languages, the main difficulty of lexical segmentation lies in the recognition of multiword expressions (MWEs). In this paper we address the following question: how useful are dependency representations for shallow MWE recognition[1] as compared to commonly used sequential labelers? We propose four different annotation schemes for lexical segmentation based on dependencies between tokens and we evaluate them on four corpora varying in sizes, languages and richness of annotation.

The originality of this approach is that dependency relations are not used for syntactic parsing nor joint lexical and syntactic parsing, but exclusively for *lexical segmentation*.

We limit our study to supervised settings, where resources for training are composed of (i) a training corpus (with MWE annotations), (ii) predicted POS tags and lemmas and (iii) information from MWE lexicons. The proposition is evaluated on three languages: English, French and Hungarian.

2 Related work

Most supervised approaches to MWE recognition focus on shallow segmentation[2]. State-of-the-art results are achieved by using sequential labelers like Conditional Random Fields (Lafferty et al., 2001), not only for contiguous MWEs (Vincze et al., 2011a; Constant et al., 2012), but also for discontiguous ones (Schneider et al., 2014a). In order to improve accuracy and linguistic representation, researchers have tried to perform joint MWE analysis and syntactic parsing using dependency parsers (Vincze et al., 2013; Candito and Constant, 2014).

Even though works on deeper MWE recognition are less common, we can cite the work of (Schneider et al., 2014a) who made a first step toward deeper lexical segmentation by performing a binary classification of idiomatic MWEs ('strong MWE') and collocations ('weak MWE').

Finally, as the MWE recognition task resembles word segmentation for languages like Chinese, we can also relate this work to approaches using dependency relations to represent word segmentation in a dependency parser (Zhao, 2009; Zhang

[1] Shallow MWE recognition corresponds to locating MWE limits. It includes the recognition of discontiguous MWEs.

[2] Shallow segmentation can be opposed to deep segmentation which includes a hierarchical representation of MWEs: e.g. lexical structuration *((Los Angeles) Lakers)*

Proceedings of the 6th Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2015), pages 1–6,
Bilbao, Spain, July 23rd 2015.

et al., 2014).

3 Lexical representation: IOB sequences vs. dependency trees

One of our objectives is to compare the performances of two types of lexical segmenters: (1) segmenters based on sequential labeling, (2) segmenters based on dependency parsing. In this section, we present these two approaches.

3.1 IOB sequential representation

We first describe the commonly used sequential representation for MWEs using the IOB tagset. The IOB annotation scheme is very popular for various segmentation tasks performed with statistical sequential labelers. It has been successfully applied to contiguous MWE segmentation like in (Vincze et al., 2011a; Constant et al., 2012). Given a sentence $w = w_1, w_2, ..., w_n$, the goal is to find the highest scoring sequence t of tags $t_1, t_2, ..., t_n$. Each tag t_i may have three values[3]: **O** marks the simple lexical units formed of a single word (i.e. not included in any MWE), **B** marks the starting word of an MWE, and **I** marks the non-starting words of an MWE. This representation is also valid for discontiguous (or gappy) MWEs, but only for the case where no MWE can be inserted in a gap. Recently, Schneider et al. (2014a) have developed an extension in order to better deal with this issue, offering the possibility to insert MWEs in gaps. For this purpose, they added two tags to the IOB tagset: **b** marks the starting word of a MWE inserted in an other MWE, and **i** marks the non-starting words of a MWE inserted in an other MWE.

For instance, in the following example, the MWE *have experience* is discontiguous and its gap contains another MWE *a bit*.

> *I/O have/B a/b bit/i of/o experience/I watching/O the/O usual/O assembly/B line/I* (taken from the CMWE corpus (Schneider et al., 2014b))

Although this IOB extension is useful for English in practice (Schneider et al., 2014a), it has theoretical limitations, as it cannot include (unbounded) recursion in MWE inserts. Note also that Schneider et al. (2014a) propose a tagset to distinguish strong and weak MWEs (cf. related work section).

[3]These 3 values can be enriched with additional information, like POS tags for instance.

3.2 Dependency tree representation

In this part, we present 4 dependency tree representations for lexical segmentation (to be performed by any off-the-shelf dependency parser). Let $w = w_1, w_2, ..., w_n$ be a sentence composed of n tokens. The segmentation of w is represented by a tree $\mathcal{T}$ that is formed of $n+1$ nodes, one root node and one node for each word.

It includes a set $\mathcal{A}$ of arcs: an arc $x \xrightarrow{l} y$ is composed of a source node x, a label l and a destination node y.

3.2.1 Representation of lexical units

A lexical unit is either a simple word (i.e. a token that is not included in an MWE) or an MWE. A lexical unit is a subtree of the lexical segmentation tree. In case of a simple word, the subtree is limited to a single node. In the case of an MWE, there exist various subtree representations in the literature, either shallow ones (Nivre and Nilsson, 2004; Eryiğit et al., 2011; Seddah et al., 2013) or deeper ones (i.e. including syntactic structure) (Vincze et al., 2013; Candito and Constant, 2014). As we solely focus on lexical segmentation (independently of any syntactic structure), we consider only shallow represention. We specifically investigate two of them:

1. **Chained representation:** MWE components are sequentially linked together like in (Nivre and Nilsson, 2004): for each consecutive word pairs (w_i,w_j), with $i < j$, within an MWE, there exists an arc $w_i \xrightarrow{MWE} w_j$.
2. **Non-chained representation:** the first MWE component is linked to every other component of the MWE like in (Seddah et al., 2013): given the first word w_i of an MWE, there exists $w_i \xrightarrow{MWE} w_j$ for each non-first word w_j of the MWE.

The root of the MWE subtree is therefore the first word of the MWE. Internal dependency arcs are labeled *MWE*. Figure 1 displays the two possible subtrees for the MWE *give a try* in the sentence *I decided to give him a try* taken from the CMWE corpus (Schneider et al., 2014b). From now on, we call *internal dependencies*, the dependencies of the MWE subtrees.

3.2.2 Representation of the segmentation

After lexical unit subtrees are built with their corresponding *internal dependencies*, it is necessary

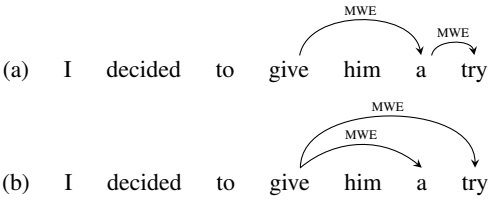

Figure 1: Internal dependencies: (a) Chained MWE representation; (b) non-chained MWE representation.

to connect them inside a valid tree[4]: we call these additional dependencies *external dependencies*. We investigate two types of representations:

1. **Chained representation:** Lexical units are sequentially linked together with a dependency arc: for each pair of consecutive lexical units represented by their roots (w_i,w_j), $i < j$, there exists an arc $w_i \xrightarrow{LEX} w_j$.

2. **Non-chained representation:** there is a dependency arc from the root node to each lexical unit: for each lexical unit represented by its root w_i, there exists an arc $root \xrightarrow{LEX} w_i$ in $\mathcal{A}$.

Figures 2 and 3 display examples of chained external dependencies and non-chained ones (respectively combined with non-chained internal dependencies and chained ones). The sentence *The staff leaves a lot to be desired*, taken again from the CMWE corpus (Schneider et al., 2014b), contains two MWEs *leaves to be desired* and *a lot*. Thus, combining the two types of internal dependencies and the two types of external ones offers four overall representations for the lexical segmentation. In the case of the non-chained external dependencies, the resulting tree is non-projective when a discontiguous MWE occurs. Otherwise trees are projective.

4 Experiments

4.1 Data sets

In this part, we present the four data sets we used to train and evaluate our lexical segmenters: the Wiki50 corpus (Vincze et al., 2011b) and the Comprehensive Multiword Expression corpus (Schneider et al., 2014a) [CMWE] for English, the

[4]A tree is valid if it has a root node and, for each word node, there exists a single path going from the root node to it.

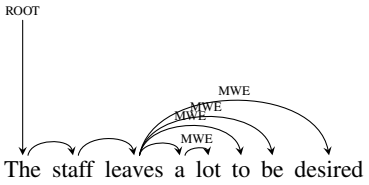

Figure 2: Chained external dependencies, combined with non-chained internal dependencies.

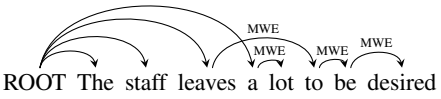

Figure 3: Non-chained external dependencies combined with chained internal dependencies.

French treebank (Abeillé et al., 2003; Seddah et al., 2013) [FTB] for French, and the Szeged treebank (Vincze et al., 2010)[Szeged] for Hungarian.

The data sets are briefly described in table 4.1. CMWE and Szeged are the only data sets including discontiguous MWEs. CMWE contains comprehensive annotations of all types of MWEs (including discontiguous ones), but is relatively small. Wiki50 and FTB are larger and contain compounds and named entities, but they do not include discontiguous MWEs. Finally Szeged is limited to Light Verb Constructions [LVC] which can be discontiguous. It is the largest corpus, but as LVCs have extremely low frequency in the text, using it to train a good system is very challenging.

Language	English		French	Hungarian
Corpus	CMWE	Wiki50	FTB	Szeged
# words	55,577	114,335	564,798	1,318,501
# MWEs	3,403	7,490	29,827	3,342
ratio	0.06	0.06	0.05	0.003

Table 1: Data sets

Szeged and Wiki50 are split using 80% for the train set and 20% for the test set. For FTB and CMWE we used the official splits. We built smaller versions of some data sets (Wiki50 and FTB) to have comparable sizes for training as compared with the CMWE[5]. We also built a smaller version for Szeged. Nonetheless, given the low frequency of MWEs, the trained model obtained near to zero accuracy scores. We therefore ommitted it in the results. Finally, we pro-

[5]For each dataset, we extracted the n first sentences, such that the number of words is the closest to the number of words in CMWE.

vided POS to all lexical segmenters[6]. We also provided predicted lemmas and information from MWE lexicons to all datasets, but the Szeged one. We used external MWE resources for English (the freely available MWE lexicons of (Schneider et al., 2014a)) and for French (the freely available MWE lexicons of (Candito and Constant, 2014)).

4.2 Sequential labeling

In this section, we provide results of sequential labeling systems on our datasets. For CMWE, we reported the state-of-the-art scores of the perceptron-based linear model[7] of (Schneider et al., 2014a), using features based on predicted POS, predicted lemmas and external MWE lexicons. For the other datasets, we ran *Wapiti* (Lavergne et al., 2010) to build and test CRF models. By using the Viterbi algorithm, it has a linear complexity in the size of the sentence and a quadratic complexity in the size of the tagset. We used the same set of features as the one defined in (Constant et al., 2012) showing state-of-the-art results on the FTB for instance (Constant et al., 2013). Results are displayed in table 4.2.

Corpus	Recall	Precision	F-score
CMWE*	48.3	61.0	53.9
Wiki50	57.5	81.7	67.5
Wiki50-short	50.4	81.7	62.3
FTB	77.6	83.3	80.4
FTB-short	63.8	80.1	71.1
Szeged	30.0	66.8	41.4

Table 2: MWE recognition results for sequence labeling on test sets. Suffix *-short* indicates CMWE-size training corpus. Symbol * indicates scores provided by (Schneider et al., 2014a)

4.3 Dependency parsing

For our experiments of lexical segmentation with dependency parsing, we used *TurboParser* (Martins et al., 2013). We chose a parser able to return non-projective trees and whose scoring scheme is rich enough to take into account sibling (for not-chained configurations) and grand-parent relations (for chained configurations). More precisely, we trained the software using the default settings, where it implements an approximate non-projective second-order parser taking into account consecutive sibling and grand-parent relations[8]. Like the sequence labeler, the parser is given predicted POS and lemmas. Results [9] are displayed in table 4.3.

Corpus	Chained external	Chained internal	Rec.	Prec.	F-score
CMWE	-	-	44.9	**65.4**	**53.3**
	-	+	45.1	64.4	53.1
	+	-	43.9	60.1	50.7
	+	+	**45.4**	56.9	50.5
Wiki50	-	-	62.5	**77.4**	**69.2**
	-	+	62.6	75.2	68.3
	+	-	63.7	74.0	68.5
	+	+	**65.7**	72.7	69.0
wiki50-short	-	-	60.6	**76.3**	**67.5**
	-	+	60.9	75.0	67.2
	+	-	61.2	74.0	67.0
	+	+	**63.0**	71.4	66.9
FTB	-	-	76.7	**79.2**	77.9
	-	+	**77.5**	78.9	**78.2**
	+	-	75.6	73.2	74.4
	+	+	75.7	72.1	73.9
FTB-short	-	-	68.1	**72.7**	**70.3**
	-	+	68.4	70.7	69.5
	+	-	67.6	65.8	66.7
	+	+	**69.3**	64.9	67.0
Szeged	-	-	**38.4**	68.9	**49.3**
	-	+	37.1	66.7	47.6
	+	-	34.2	70.1	46.0
	+	+	35.6	**71.8**	47.6

Table 3: Test results with TurboParser

5 Discussion

When comparing the two approaches on all datasets, we reach varying results. First of all, we can notice that the dependency parsing system outperforms the sequential labelling one on Wiki50 and Szeged. For instance, on Szeged the best parsing system outperforms the CRF-based system by around 8 points. The MWEs annotated in this corpus are restricted to LVCs which often exhibit discontinuity. Then, the two systems reach on-par scores on the CMWE. Finally, we can notice that the parsing system obtains disappointing results on FTB: it is overall lower than the sequential labeling one by more than 2 points.

Among the systems based on dependency parsing, we can observe that the one us-

[6]We use the Stanford POS tagger (Toutanova et al., 2003) for CMWE and Wiki50. For the FTB and Szeged, we used the predicted POS provided respectively in the SPMRL data set (Seddah et al., 2013)via jackknifing and by the authors of Szeged.

[7]We did not report the scores of the system including cluster-based features to be fairly comparable with the dependency parsing systems.

[8]Higher-order non-projective parsing is an NP-complete problem. However, the concrete complexity of TurboParser is difficult to analyze since it relies on decomposition methods. The most complex factor (implementing the spanning tree) has quadratic time complexity. But, as factors need to agree on partial structures, this calculation might be carried out many times in pathological cases.

[9]We also ran experiments with MaltOptimizer (Ballesteros and Nivre, 2012) [resp. *Mate* parser (Bohnet, 2010)] that show lower [resp. comparable] results than [resp. with] *TurboParser*.

ing non-chained external dependencies combined with non-chained internal dependencies, always reaches the best results, except for FTB (rank 2). It has very high precision scores as compared with the others, while recall is often slightly lower. It seems that sibling-based features capture more relevant information for lexical segmentation than grand-parent ones for MWEs of more than two tokens.

We can note that results on Szeged and CMWE are overall lower than the ones on the other datasets (cf. results of systems trained on datasets of comparable size (suffix *short*)). This can be explained by the fact that the two first datasets contain discontiguous MWEs, which are more difficult to predict as it requires more structural information. Furthermore, the Hungarian system do not include lexicon-based feature, which could also explain the lower scores on the Szeged corpus, in addition with the low frequency of LVCs.

6 Conclusions and future work

This paper presents a novel representation for lexical segmentations as trees over tokens that can adequately model gappy and interleaved MWEs without any restriction on the depth of the hierarchical structures. Using an off-the-shelf dependency parser, we were able to recover results with near state-of-the-art accuracy. Future work will focus on dependency representations for deeper, i.e. nested, MWE recognition (ex. *(make a (big deal))*). We would like to exploit tree representations in order to design models for joint parsing and lexical segmentation using trees for both dimensions. Finally we believe that similar representations may be useful word tokenization, especially in Morphologically Rich Languages where token frontiers are ambiguous.

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. Building a treebank for French. In Anne Abeillé, editor, *Treebanks*. Kluwer, Dordrecht.

Miguel Ballesteros and Joakim Nivre. 2012. MaltOptimizer: An Optimization Tool for MaltParser. In *Proceedings of the System Demonstration Session of the Thirteenth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

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

Marie Candito and Matthieu Constant. 2014. Strategies for Contiguous Multiword Expression Analysis and Dependency Parsing. In ACL, editor, *ACL 14 - The 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, United States, June.

Matthieu Constant, Anthony Sigogne, and Patrick Watrin. 2012. Discriminative strategies to integrate multiword expression recognition and parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, pages 204–212.

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.

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.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 504–513.

André F. T. Martins, Miguel B. Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 617–622. The Association for Computer Linguistics.

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.

Nathan Schneider, Emily Danchik, Chris Dyer, and Noah A. Smith. 2014a. Discriminative lexical semantic segmentation with gaps: running the MWE gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206, April.

Nathan Schneider, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T. Mordowanec, Henrietta

Conrad, and Noah A. Smith. 2014b. Comprehensive annotation of multiword expressions in a social web corpus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 455–461, Reykjavík, Iceland, May. ELRA.

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*, Seattle, WA.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259.

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

Veronica Vincze, István Nagy, and Gábor Berend. 2011a. Detecting noun compounds and light verb constructions: a contrastive study. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE'11)*, pages 116–121.

Veronica Vincze, István Nagy, and Gábor Berend. 2011b. Multiword expressions and named entities in the wiki50 corpus. In *Proceedings of the conference on Recent Advances in Natural Language Processing (RANLP'11)*, pages 289–295.

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.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Character-level chinese dependency parsing. In *Proceedings of the 52th Conference of the Association for Computational Linguistics*. Association for Computational Linguistics.

Hai Zhao. 2009. Character-level dependencies in chinese: usefulness and learning. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 879–887. Association for Computational Linguistics.