PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 99 APRIL 2013 59–75

Accounting for Contiguous Multiword Expressions in Shallow Parsing

Matthieu Constant[a], Olivier Blanc[b], Patrick Watrin[b]

[a] Université Paris-Est, LIGM, CNRS, France
[b] Université de Louvain, CENTAL, Belgique

Abstract

In this paper, we focus on chunking including contiguous multiword expression recognition, namely *super-chunking*. In particular, we present different strategies to improve a super-chunker based on Conditional Random Fields by combining it with a finite-state symbolic super-chunker driven by lexical and grammatical resources. We display a substantial gain of 7.6 points in terms of overall accuracy.

1. Introduction

Multiword expressions (MWEs) are key parts of Natural Language Processing (Sag et al., 2002). But they have long been neglected in empirical parsing researches. Preliminary works like Nivre and Nilsson (2004); Arun and Keller (2005) integrated such expressions in parsers but by considering a gold MWE recognition. In recent pioneer studies, realistic MWE recognition has started to be integrated in shallow parsers (Korkontzelos and Manandhar, 2010) and in deep parsers (Cafferkey et al., 2007; Green et al., 2011; Constant et al., 2012).

In this paper, we focus on shallow parsing, more precisely on chunking including multiword expression recognition, namely *super-chunking* (Blanc et al., 2007). The considered MWEs are contiguous and consist of compounds, nominal collocations, multiword terms, named entities like dates, person and organization names, etc. Historically, chunking can conveniently be implemented with cascades of finite-state transducers (Abney, 1996; Joshi and Hopely, 1997; Ait-Mokhtar and Chanod, 1997; Nasr and Volanschi, 2005). Blanc et al. (2007) successfully extended this finite-state frame-

Corresponding author: mconstan@univ-mlv.fr
Cite as: Matthieu Constant, Olivier Blanc, Patrick Watrin. Accounting for Contiguous Multiword Expressions in Shallow Parsing. The Prague Bulletin of Mathematical Linguistics No. 99, 2013, pp. 59–75.
doi: 10.2478/pralin-2013-0004.

work to *super-chunking*. They implemented a super-chunker, namely POM, mainly driven by large-scale lexical and grammatical finite-state resources. Shallow parsing can also be seen as a sequence annotation task (Ramshaw and Marcus, 1995) that is very well modeled by discriminative models (e.g. Kudo and Matsumoto, 2001; Sha and Pereira, 2003; Tsuruoka et al., 2009).

In this paper, we consider the use of Conditional Random Fields (CRF). Such approach relies on a reference annotated corpus. But, what happens if this corpus does not entirely comply with our needs in terms of super-chunk annotations? Especially, what happens if some expected types of MWEs are not encoded? In the following, we propose different strategies to adapt a CRF-based super-chunker to our needs by combining it with a symbolic super-chunker driven by lexical and grammatical resources (like POM). The proposed solutions are all evaluated on French as many MWE resources are available for this language.

The paper is organized as follows. In Section 2, we define super-chunking and our target annotations. Section 3 is devoted to the description of the available resources. Then, in Section 4 we present a simple CRF-based super-chunker that we consider our *baseline*. Next, we detail POM architecture (Section 5). Finally, we present two combination solutions (Section 6) and evaluate them (Section 7).

2. Super-chunking

2.1. Super-chunks

Super-chunks are non-recursive syntactic constituents, like standard chunks (Abney, 1991), with the difference that they can contain multiword expressions, as we defined it in Blanc et al. (2007). For instance, *marge d'exploitation* (trading margin) is considered a compound noun, so the utterance *la marge d'exploitation* (the trading margin) is annotated as a nominal super-chunk (XN), while standard chunking would have produced a sequence of a noun phrase (XN) followed by a prepositional phrase (XP).[1] Considering super-chunks instead of standard chunks has two main interests: (1) it reduces combinatorial complexity for shallow parsing because some ambiguities are resolved with MWE recognition[2] (Korkontzelos and Manandhar, 2010) ; (2) it allows for the identification of semantic units as MWEs form idiomatic units (Baldwin and Nam, 2010).

[1] The utterance would be annotated in standard chunks like below:

[XN la marge] [XP d'exploitation]
(the trading margin)

[2] Korkontzelos and Manandhar (2010) showed that recognizing MWEs could improve chunking. They limited the experiment to some types of MWEs that do not change the chunk annotation definition.

Example 1 illustrates super-chunking.[3] The annotated sentence contains 4 MWEs (between brackets): the adverbial time expression *durant le premier trimestre 2007* (during the first trimester 2007), the nominal determiner *l'ensemble des* (the whole), the noun *chiffre d'affaires brut* (gross sales) and the complex numerical determiner *6 121 millions de*. It is composed of a sequence of 6 super-chunks (instead of 11 chunks with standard chunking).

(1) *[(Durant le premier trimestre 2007)], [(l'ensemble des) activités] [au Maroc] [ont généré] [un (chiffre d'affaires brut)] [de (6 121 millions de) dirhams]*
([During the first trimester 2007], [the whole activities] [in Marocco] [generated] [gross sales] [of 6,121 million dirhams].)

Several multiword expressions can be combined into a same super-chunk because any lexical item can be a multiword expression. For instance, let's consider the following annotated sequence:

[XN La température] [XP (à l'intérieur de) (beaucoup de) maisons] [XP en Moldavie]
(*[XN the temperature] [XP inside a lot of houses] [XP in Moldavia]*)

The whole phrase *à l'intérieur de beaucoup de maisons* is considered to be a prepositional super-chunk (XP) because *à l'intérieur de* (inside) is a multiword preposition, *beaucoup de* (a lot of) is a multiword determiner and *maisons* (houses), a simple noun.

Verbal chunks are also very specific because they can include auxiliaries in the sense of Gross (1999), inserts, clitics and negation. For example, the sentence

Jean n'a pas pu les trouver
(John could not find them)

is annotated:

[XN Jean] [XV n'a pas pu les trouver]
([John] [could not find them])

The discontinuous sequence *n'... pas* (not) is a negation, *a ... pu* is the preterit form of the modal verb *pouvoir* (to can) and *les* is an accusative clitic.

2.2. Target annotation

The annotation tagset of the super-chunker is given in Table 1. Lexical items that do not belong to any chunk like conjunctions, punctuations or relative pronouns are labelled with the tag O (meaning other). It is possible to have multiword O like for multiword conjunctions (e.g. *bien que* – although). In that case, the whole unit is annotated O: *bien_que/O*. Example 1 is then fully annotated as provided in Table 2.

The target evaluation corpus is a mix of a part of the novel "Le Tour du Monde en 80 jours" (Around the World in 80 days) by Jules Verne, as well as some reports of French

[3] This example shows the super-chunk segmentation without providing the labels.

TAG	DESCRIPTION
XA	adjectival chunk
XADV	adverbial chunk
XN	nominal chunk
XP	prepositional nominal chunk
XV	verbal chunk
XVP	prepositional verbal chunk

Table 1. Super-chunker tagset

CHUNK	TAG
Durant le premier trimestre 2007	XADV
,	O
l'ensemble des activités	XN
au Maroc	XP
ont généré	XV
un chiffre d'affaires brut	XN
de 6 121 millions de dirhams	XP

Table 2. Chunking result example

Parliament sessions. It is composed of 424 sentences, 8,319 tokens and 4,394 super-chunks. It has been semi-automatically annotated by T. Nakamura and S. Voyatzi at the LIGM laboratory. Both annotators have independently validated the whole corpus. In case of disagreement, a final decision was made after discussion. The super-chunk annotations exactly match the target annotations. Despite its small size, this corpus is therefore adequate for the evaluation of our tool. It is much harder to parse than a journalistic corpus because it contains specific terms (Parliament report section) and very long named entities (Around the World in 80 days). It is also full of dialogs. We assume that we have no development corpus of the same type. This means that the super-chunking process is completely blind.

3. Resources

In this section, we present the resources available to train and tune our super-chunker. They are composed of a treebank and lexical resources. The chunk annotations directly derived from the treebank do not exactly match our target ones.

3.1. Annotated corpus

The French Treebank (FTB) is a syntactically annotated corpus made of journalistic articles (Abeillé et al., 2003). Our edition comprises 584,987 tokens and 19,108 sentences. One benefit of this corpus is that compounds are marked. Their annotation was driven by linguistic criteria such as the ones in Gross (1986). Some organization and location names are also encoded, e.g. *Royal Zenith of Great Neck*, *San Francisco*. In total, around 5.6% of all lexical units are marked as multiword expressions. Some types of MWEs are missing like nominal collocations,[4] time expressions (date, duration, etc.), person names, etc.

We automatically converted the treebank to super-chunks in the target chunk tagset. In most cases, the chunk definition corresponds to the expected ones, except for verbal chunks when the verbs are combined with auxiliaries and modal verbs. For instance, the sentence *Marie peut changer* (Marie can change) is annotated *(XN Marie) (XV peut) (XV changer)* in the French Treebank, whereas the expected annotation is *(XN Marie) (XV peut changer)* because *peut* (can) is a modal verb.

We split the corpus in two distinct sections: a training section (90%) and a development section (10%). The training section was used to learn the CRF model, whereas the development section was used for the tuning of the features, the lexical and grammatical resources and the super-chunker POM.

3.2. Lexical resources

The lexical resources include large-scale dictionaries developed by linguists. They are lists of lexical entries, each of them being composed of an inflected form, a lemma, a part-of-speech (POS), morphological information (*e.g.* gender, number), syntactic information (*e.g.* transitive or intransitive verbs) and semantic information (*e.g.* human feature for nouns). They encode not only simple words but also multiword expressions like compounds. They are compressed in the form of FSTs in order to be efficiently applied to the text.

All dictionaries that we used are listed in Table 3. The larger ones were developed between the mid-80's and the mid-90's by linguists at the University of Paris 7: DELAF (Courtois, 1990) is composed of 746,198 inflected simple forms; DELACF (Courtois

[4] Collocations are combinations of words that co-occur more often than by chance. They are usually defined through statistical criteria.

et al., 1997) contains 255,163 inflected compounds (mostly composed of compound nouns). Compounds are of the following types :

- nouns: *pomme de terre* (potato), *faux témoignage* (perjury)
- prepositions: *au milieu de* (in the middle of), *à cause de* (because of)
- adverbs: *par ailleurs* (moreover), *en pratique* (in practice)
- conjunctions: *bien que* (although), *pendant que* (while)

The dictionary PROLEX (Piton et al., 1999) is a list of toponyms used in order to recognize location names. There exist several additional dictionaries containing organizations, first names, domain-specific terms and additional lexical entries found during the development process.

Name	Description	#entries	Reference
DELAF	Simple words	948,177	(Courtois, 1990)
DELACF	Compound words	255,163	(Courtois et al., 1997)
PROLEX	Toponyms	192,249	(Piton et al., 1999)
MISC	Additional dictionaries	34,151	

Table 3. Dictionaries

Our lexical resources also contain a library of strongly lexicalized local grammars. Local grammars (Gross, 1997) are Recursive Transition Networks (RTNs) (Woods, 1970) and theoretically recognize algebraic languages. They are of great interest for representing local lexical and syntactic constraints in a simple and compact way. We use them mostly to describe MWEs. They can define syntactic classes such as noun determiners and even syntactico-semantic classes such as time adverbials. Linguistic descriptions are in the form of Finite-State Graphs (Silberztein, 1994) on an alphabet made of terminal and non-terminal symbols. A terminal symbol is a word or a lexical mask. A lexical mask is an underspecified lexical entry (some features are missing) equivalent to a feature structure representing a set of lexical entries: *e.g.* the lexical mask <*noun+plural*> matches all nouns in the plural. Finally, a non-terminal symbol is a reference to another graph. A graph represents a transducer and its output is the annotation assigned to utterances described in the graph. An example of a local grammar is given in Figure 1.[5] This grammar describes time adverbials and recognizes structures like *en mars 2007* (in March 2007) and *cinq minutes plus tard* (five minutes later). The sequences recognized by this graph are tagged as time adverbs (`ADV+time`). Strings between < and > are lexical masks: for instance, <*minute*> stands for the inflected forms whose lemma is *minute*. Greyed vertices are call to other graphs. For ex-

[5] The local grammars are drawn using the graph editor of the Unitex platform (Paumier, 2003).

ample, `Dnum` and `month` are graphs that recognize numerical determiners and month names.

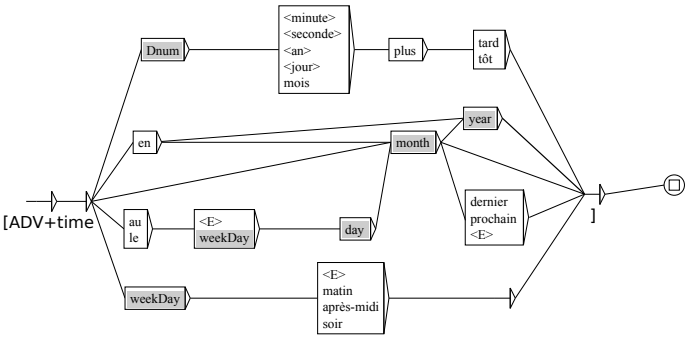

Figure 1. Local grammar of time adverbials

Practically, our lexical resources include a network of 302 graphs (in total, 2,853 states and 11,425 transitions in the RTN representation). Local grammars recognize sequences of the following types:

- nouns: function names [*ministre anglais de l'Agriculture* (English minister of Agriculture)], location names [*lac des Bois* (lake of the Woods)] and person names [*M. John Smith*]
- prepositions: locative prepositions [*à dix kilomètres au nord de* (ten kilometers north of)]
- determiners: numerical determiners [*vingt-sept* (twenty seven), *des milliers de* (thousands of)], noun determiners [*dix grammes de* (ten grams of)]
- adverbs: time adverbials [*en octobre 2006* (in October 2006)]

Local grammars identifying named entities (i.e. time adverbials and organization, location and person names) were partly constructed from Martineau et al. (2009). Graphs recognizing noun determiners come from Silberztein (2003); Laporte (2007). All of them are compiled into equivalent FSTs[6] (see Table 4).

4. Super-chunking with Conditional Random Fields

Chunking, and by extension super-chunking, can be seen as a sequential annotation task (Ramshaw and Marcus, 1995). Each word is assigned a tag Y-TAG: TAG is the tag of the chunk it belongs to and Y indicates its relative position in the chunk (*B* for beginning; *I* for the remaining positions). Table 5 shows an example of a chunked

[6]The local grammar representing determiners *det* is strictly recursive and there is no equivalent FST (just an approximation).

type	#graphs	#states in RTN	#transitions in RTN	#states in equiv. FST	#transitions in equiv. FST
adv	56	594	232	1,722	33,971
det	180	1718	746	/	/
noun	24	146	414	112	573
prep	43	386	1,215	2,599	62,657

Table 4. Compilation of local grammars into equivalent FSTs

text using this annotation scheme: column CHUNK corresponds to the output chunk label.

WORD	MWE+POS	CHUNK
l'	B-DET	B-XN
ensemble	I-DET	I-XN
des	I-DET	I-XN
activités	B-N	I-XN
ont	B-V	B-XV
généré	B-V	I-XV
121	B-DET	B-XN
millions	I-DET	I-XN
de	I-DET	I-XN
dirhams	B-N	I-XN

Table 5. Chunking result example

Several studies (Sha and Pereira, 2003; Tsuruoka et al., 2009) have shown that Linear-chain Conditional Random Fields (LCRF) are very competitive for chunking. The models usually incorporate features computed from predicted POS and the words themselves like in Tsuruoka et al. (2009). Nevertheless, standard chunkers do not account for MWE recognition. In Constant and Tellier (2012), we showed that MWE recognition could be successfully performed jointly with POS tagging. The column MWE+POS in Table 5 corresponds to the output of such joint task. A first super-chunking strategy is therefore to use such MWE-aware labels instead of simple POS. The model is trained on the training corpus *as is*, although chunk annotations do not entirely match our target ones. We use a similar set of feature templates as Tsuruoka

et al. (2009) for their base chunker, except that we deal with MWE-aware POS labels instead of simple ones: word unigrams, bigrams and trigrams; MWE+POS unigrams, bigrams and trigrams. We trained[7] two MWE-aware POS tagger models using features defined in Constant and Tellier (2012): the first model (called WITH) contains all features including features based on data available in the external lexical resources (namely lexicon-based features); the second one (called WITHOUT) contains all features but the lexicon-based ones. In order to select the best model for chunking, we applied the two models on the FTB development corpus. Results are provided in Table 6. The upper part corresponds to overall scores: MWE+POS is the joint MWE recognition and POS tagging accuracy in terms of labeled F_1-measure; U_1 stands for the chunking unlabeled F_1-measure indicating the super-chunking segmentation accuracy; F_1 stands for the chunking labeled F_1-measure. The lower part details the F_1-measure for each chunk label. The column *#Chunks* indicates the number of chunks for each label in the development corpus.

	#Chunks	WITHOUT	WITH
MWE+POS	–	93.9	**94.5**
U_1	–	**92.1**	91.2
F_1	–	**90.1**	88.8
O	10,827	97.4	97.0
XA	2,506	81.3	79.3
XADV	1,854	81.4	77.0
XN	7,161	86.6	85.6
XP	8,397	86.9	85.2
XV	5,377	91.5	90.7
XVP	788	92.3	90.7

Table 6. Baseline results on the FTB development section

The best super-chunker reaches around 90% accuracy on the development corpus. We observe that chunk segmentation costs around 8 points[8]. We consider it our *baseline*. Surprisingly, the best super-chunker uses a MWE-aware POS tagger including no lexicon-based features, whereas joint MWE and POS labeling is much better with lexicon-based features. This might show that errors caused by the lexicon-based tagger are critical and cause much more damages for super-chunking.

[7] We trained the CRF models by using the software Wapiti (Lavergne et al., 2010), with the same settings as in (Constant and Tellier, 2012).

[8] The chunk segmentation cost is $100 - U_1$.

5. Super-chunking with a finite-state lexicon-driven approach

Blanc et al. (2007) proposed a finite-state architecture to handle super-chunking, that was developed in the tool POM. It is based on a cascade of finite-state transducers (FSTs), similarly to the historical finite-state approach of shallow parsing (Joshi and Hopely, 1997; Abney, 1996; Ait-Mokhtar and Chanod, 1997). It relies on external large-coverage lexical resources, as in Silberztein (1994). The chunker is composed of three successive stages as illustrated in the diagram in Figure 2: (1) an enhanced ambiguous lexical analysis, (2) an ambiguous chunk analysis, (3) a chunk disambiguation module. The whole system is mainly driven by linguistic resources in the form of lexicons and local grammars. There might also be preprocessing and post-processing stages, for instance, to deal with disfluencies in speech transcripts (Blanc et al., 2010). In this paper, we used the same super-chunker architecture.

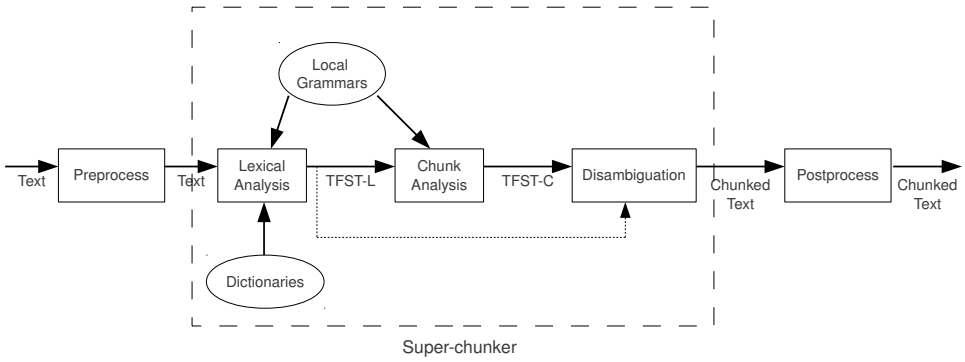

Figure 2. Process diagram

The lexical analysis module takes as input a text segmented into sentences and tokens. It uses large-coverage lexical resources in the form of morphosyntactic dictionaries and lexicalized local grammars, all compiled into FSTs (cf. Section 3). These resources are applied iteratively to the text. The module generates an acyclic text finite-state transducer (TFST-L) representing lexical ambiguities for simple words and multiword expressions for each sentence of the input text. First, a dictionary lookup associates each token with all its possible morphosyntactic tags and recognizes MWEs. The output of the lookup is a finite state transducer (TFST). Then, a cascade of strongly lexicalized grammars is iteratively applied to the TFST, which is augmented with the analyses of the matching MWEs.

Chunk analysis is also based on a cascade of finite transducers applied to TFST-L, which is augmented each time a new chunk is found. It returns the text finite trans-

ducer TFST-C. The module is composed of 13 successive stages, each stage corresponding to the application of a given FST recognizing a type of syntactic structure. We successively identify :

1. adverbials (XADV): simple adverbs or multiword adverbials that have been recognized during the lexical analysis
2. adjectival chunks (XA): adjectives that can be preceded by an adverb
3. nominal chunks (XN): simple noun phrases, named entities, some types of pronouns
4. prepositional chunks (XP): XN preceded by a preposition
5. verbal chunks (cascade of 9 FSTs): passive and active forms of infinitive, past participle, gerund and simple verbal chunks, complex structures integrating auxiliaries in the sense of Gross (1999).

The syntactic patterns were constructed manually in the form of local grammars constituting a global network of 115 graphs (1,109 states and 6,054 transitions in the RTN representation). Then, they were compiled into equivalent FSTs which comprise 823 states and 10,703 transitions in total. Our FSTs represent not only purely syntactic patterns (e.g. XN can be composed of a noun preceded by a determiner) but also lexico-syntactic patterns. For example, we used the lexico-syntactic patterns to:

- describe auxiliaries in the sense of Gross (1999) followed by a verb in the infinitive, such as *viser à* (to aim at), *avoir peur de* (to be afraid of)
- describe fixed XNs such as *les uns et les autres* (one and every)
- describe intensive adverbials that can modify adjectives such as *très* (very), *un peu* (a little).

The generated TFST-C is composed of both POS and chunk tags found in the lexical and chunk analyses. In order to remove ambiguity, the chunker includes an incremental disambiguation module removing paths until total linearization is reached. Blanc et al. (2007) proposed the three following stages: disambiguation with handcrafted rules (given an ambiguity and a context, selection of a tag by removing the transitions corresponding to the other tags); algorithm keeping the shortest paths (in order to favor multiword analyses); then a simple statistical linearization (based on the probability to associate the tag of the chunk with its head word).

In this paper, we developed a simpler disambiguation stage. It was limited to one module: the application of the shortest path algorithm on TFST-C, which was shown to be the best. The TFST-C weighting is manually set, giving priority to chunk tags. This lighter disambiguation module forced us to use a specific simple word dictionary with very few ambiguities. To do so, we applied a standard POS tagger[9] on simple words at the preprocessing stage. From it, we built a dictionary of simple words that we applied at the lexical stage. We also applied all MWE lexical resources described above. The standard POS tagger model integrates features based on all our lexical resources (cf. Section 3).

[9] We used *lgtagger* (Constant and Tellier, 2012).

6. Combining CRF-based super-chunker and lexicon-driven super-chunker

Our intuition is that POM super-chunker is more accurate for segmentation than the CRF-based one (our *baseline*) because it is driven by large-coverage lexical resources full of MWEs, all compatible with our target annotation. Furthermore, the CRF-based super-chunker should be more accurate to label the identified chunk segments, as POM does not have any advanced disambiguation tools. Therefore, it sounds interesting to combine both systems.

For this purpose, we developed two different combination strategies. The first one consists in merging the outputs of both super-chunkers. It is called *merge*. The second one consists in adapting the training corpus by merging it with the POM output, and then in learning a new CRF model. It is called *adapt*. The two strategies are based on the same merging procedure. This procedure takes two annotations as input and generates a new annotation, as illustrated in Table 7. It works as follows. We first perform super-chunk segmentation by gathering the two input annotations in the same directed graph. Each node corresponds to a chunk segment at a given position. An arc links two chunks c_1 and c_2 if the end of c_1 coincides with the beginning of c_2 in the text. We find the final segmentation by applying a shortest path algorithm that favors the longest chunks, and, in case of segmentation ambiguity, the chunks found by POM. The graph computed for our example is provided in Figure 3. Once the chunk segmentation is selected, we assign to each chunk its label found in the annotations. In case of ambiguity, the CRF-based label (or the FTB reference one) is chosen, except for some specific cases like adverbials. This algorithm is very easy to implement and formulates our initial intuition. Moreover, it could be easily extended from 2 to n annotations.

POM output		FTB Reference annotation		Merge	
Le 10 juillet 'On July 10'	XADV	Le 10 juillet 'On July 10'	XN	Le 10 juillet 'On July 10'	XADV
Luc Ferry 'Luc Ferry'	XN	Luc Ferry 'Luc Ferry'	XN	Luc Ferry 'Luc Ferry'	XN
put rencontrer 'could meet'	XV	put 'could'	XV	put rencontrer 'could meet'	XV
		rencontrer 'meet'	XV		
le ministre 'the minister'	XN	le ministre des affaires sociales 'the minister for social affairs'	XN	le ministre des affaires sociales 'the minister for social affairs'	XN
des affaires sociales 'for social affairs'	XP				

Table 7. Example of two annotations for the merging procedure

For the method *adapt*, the CRF model might incorporate additional features (as compared with the *baseline* model). These features are based on the annotations generated by POM. Given a position i, let chk(i) be the POM-predicted tag of the chunk

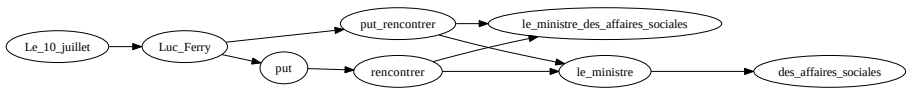

Figure 3. Example of graph for the merging procedure

the current token belongs to. $\mathrm{chkbi}(i)$ indicates the relative position of the token in the POM-predicted chunk: B for the starting position and I for the remaining positions. $c(i)$ is the current output label (a chunk tag). The additional feature templates are provided in the Table 8. We tested this strategy both with the baseline features (*adapt-base*) and with all features including the additional ones described in Table 8 (*adapt-advanced*).

$\mathrm{chk}(i+j)/\mathrm{chkbi}(i+j)/c(i)$	with $j \in [-2, 2]$
$\mathrm{chkbi}(i+j)/c(i)$	with $j \in [-2, 2]$
$\mathrm{chk}(i+j)/c(i)$	with $j \in [-2, 2]$

Table 8. Additional feature templates

7. Evaluation

This section is devoted to the evaluation of the three proposed solutions on the target MIX corpus. We compare them with the two simple super-chunkers presented in Sections 4 and 5. The results are provided in Table 9.

We can first notice that the performances of the baseline CRF-based super-chunker and, respectively, our version of POM are quite low (76%) as compared with the scores obtained on the FTB-dev (90%) and, respectively, the scores obtained on journalistic articles by the POM version described in Blanc et al. (2007) (around 95%). The difference with the results on FTB-dev can be partly explained by the fact that some types of multiword expressions are not marked in the FTB-train and some verbal super-chunks do not match with what is expected (cf. low accuracy for XV and XVP). Moreover, in Blanc et al. (2007), a great effort was made on the tuning of POM resources and disambiguation rules. We developed them with the help of a corpus of the same type as the evaluation corpus. In our case, there are no means to tune our resources. In addition, there are some disambiguation modules missing, even though this is partly

	# chunks	baseline	POM	merge	adapt base	adapt advanced
U_1	–	80.5	84.7	**87.9**	85.9	**88.2**
F_1	–	76.0	76.1	83.2	81.3	**83.6**
O	1,591	95.4	86.8	95.5	95.5	**95.8**
XA	157	58.5	60.0	66.5	62.2	**68.6**
XADV	343	49.3	48.8	59.5	56.4	**59.7**
XN	832	76.3	74.2	81.8	78.8	**82.0**
XP	591	62.6	69.8	72.2	66.4	**72.8**
XV	794	65.0	82.2	82.4	82.2	**82.8**
XVP	86	72.4	80.2	**80.5**	76.1	77.6

Table 9. Final results on the evaluation corpus MIX

compensated by the use of a POS tagger to have less ambiguous lexical resources. Nevertheless, the main cause of this performance drop is simply that the MIX corpus is hard to parse.

Furthermore, we can observe that the baseline CRF-based super-chunker and our version of POM have very comparable results. They have comparable accuracies on adjective, adverbial and noun phrases. But they show very different results on the other categories. POM is very bad for identifying non-chunks, i.e. tag O: around 9 point difference with the CRF-based. This is mainly due to the light disambiguation module of POM. The CRF-based super-chunker obtains bad results for prepositional phrases (-7 point difference with POM) and verbal chunks (at worse, -17 points). This is not surprising for verbal chunks as the training corpus is annotated with a verbal super-chunk definition different from the evaluation corpus. For the prepositional phrases, this is due to incorrect multiword preposition recognition.

We can notice that our assumptions on the performances of the baseline and POM are verified: POM is better for segmentation (84.7% vs. 80.5%), whereas the baseline is better at disambiguation (-3.5 vs. -8.6 points as compared with the segmentation scores). As expected, the combination strategies show great improvements. The strategy *adapt-advanced* reaches the best accuracy with a very substantial gain of +7.6 points as compared with the baseline. The *merge* method obtains slightly lower scores, but they are comparable. We can see that *adapt-base* has lower improvement: +5.5 points. This shows that the use of features based on the super-chunks predicted by POM are of great interest (gain of around 2 points). For non-chunks, verbal and prepositional

phrases, the combined super-chunker reaches results comparable with the ones obtained by the best simple chunker for these categories. For adjective, adverbial and noun phrases, we observe that the two super-chunkers are complementary as their combination achieves quite better scores as compared with the best simple chunker for each of these categories: +10 points for adverbials, +9 points for adjective phrases, +6 points for noun phrases.

8. Conclusions and Future Work

In this paper, we focused on shallow parsing, more precisely on chunking including MWE recognition, namely *super-chunking*. As it is sometimes hard to have a training corpus with the exact same annotations as expected, it can be easier to use a chunker driven by lexical resources that are usually adapted to one's needs. We have shown how to improve a CRF-based super-chunker by coupling it with a finite-state symbolic lexicon-driven super-chunker. We used a procedure merging two chunk annotations: either to merge the two super-chunker outputs, or to adapt the training corpus with the lexicon-driven super-chunker. In the second case, the CRF model is even improved when integrating additional features based on the lexicon-driven super-chunker. We display a substantial gain of 7.6 points in terms of general accuracy for this latter solution. Future work might consist in improving the lexicon-driven super-chunker in order to have a more precise merging procedure and have finer features for CRF models. It would be interesting to extend the evaluation to other target domains.

Bibliography

Abeillé, A., L. Clément, and F. Toussenel. Building a treebank for French. In Abeillé, A., editor, *Treebanks*. Kluwer, Dordrecht, 2003.

Abney, S. P. Parsing by chunks. In Berwick, Robert C., Steven P. Abney, and Carol Tenny, editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 257–278. Kluwer Academic Publishers, Dordrecht, 1991.

Abney, S. P. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1996.

Ait-Mokhtar, S. and J.-P. Chanod. Incremental finite-state parsing. In *Proceedings of the fifth Conference on Applied Natural Language Processing (ANLP'97)*, 1997.

Arun, A. and F. Keller. Lexicalization in crosslinguistic probabilistic parsing: The case of french. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005.

Baldwin, T. and K.S. Nam. Multiword expressions. In *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, 2010.

Blanc, O., M. Constant, and P. Watrin. Segmentation in super-chunks with a finite-state approach. In *Proceedings of the Workshop on Finite-State Methods for Natural Language Processing (FSMNLP'07)*, 2007.

Blanc, O., M. Constant, A. Dister, and P. Watrin. Partial parsing of spontaneous spoken French. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

Cafferkey, C., D. Hogan, and J. van Genabith. Multi-word units in treebank-based probabilistic parsing and generation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'07)*, 2007.

Constant, M. and I. Tellier. Evaluating the impact of external lexical resources into a crf-based multiword segmenter and part-of-speech tagger. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, 2012.

Constant, M., A. Sigogne, and P. Watrin. 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, 2012.

Courtois, B. Un système de dictionnaires électroniques pour les mots simples du français. *Langue Française*, 87:11–22, 1990.

Courtois, B., M. Garrigues, G. Gross, M. Gross, R. Jung, M. Mathieu-Colas, A. Monceaux, A. Poncet-Montange, M. Silberztein, and R. Vivés. Dictionnaire électronique DELAC : les mots composés binaires. Technical Report 56, LADL, University Paris 7, 1997.

Green, S., M.-C. de Marneffe, J. Bauer, and C. D. Manning. Multiword expression identification with tree substitution grammars: A parsing tour de force with French. In *Proceedings of the conference on Empirical Method for Natural Language Processing (EMNLP'11)*, 2011.

Gross, M. Lexicon grammar. the representation of compound words. In *Proceedings of the conference on Computational Linguistics (COLING'86)*, 1986.

Gross, M. The construction of local grammars. In Roche, E. and Y. Schabes, editors, *Finite-State Language Processing*, pages 329–352. The MIT Press, Cambridge, Mass., 1997.

Gross, M. Lemmatization of compound tenses in english. *Lingvisticae Investigationes*, 22, 1999.

Joshi, A. and P. Hopely. A parser from antiquity. *Natural Language Engineering*, 2(4), 1997.

Korkontzelos, I. and S. Manandhar. Can recognising multiword expressions improve shallow parsing ? In *Proceedings of the Conference on Human Language Technologies and the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'10)*, pages 636–644, 2010.

Kudo, T. and Y. Matsumoto. Chunking with support vector machines. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'01)*, 2001.

Laporte, E. Extension of a grammar of French determiners. In *Proceedings of the international conference on Lexicon and Grammar (LGC'07)*, pages 65 – 72, 2007.

Lavergne, T., O. Cappé, and F. Yvon. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 504–513, 2010.

Martineau, C., T. Nakamura, L. Varga, and S. Voyatzi. Annotation et normalisation des entités nommées. *Arena Romanistica*, 4:234–243, 2009.

Nasr, A. and A. Volanschi. Integrating a POS tagger and a chunker implemented as weighted finite state machines. In *Proceedings of the Workshop on Finite-State Methods and Natural Language Processing (FSMNLP'05)*, pages 167 – 178, 2005.

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

Paumier, Sébastien. *De la reconnaissance de formes linguistiques à l'analyse syntaxique*. PhD thesis, Université Paris-Est Marne-la-Vallée, 2003.

Piton, O., D. Maurel, and C. Belleil. 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)*, pages 233–237, 1999.

Ramshaw, L. A. and M. P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pages 88 – 94, 1995.

Sag, I. A., T. Baldwin, F. Bond, A. Copestake, and D. Flickinger. Multiword expressions: A pain in the neck for nlp. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing (CICLing '02)*, pages 1–15, London, UK, 2002. Springer-Verlag.

Sha, F. and F. Pereira. Shallow parsing with conditional random fields. In *Proceedings of the Conference on Human Language Technologies and the Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, pages 213 – 220, 2003.

Silberztein, M. Intex: a corpus processing system. In *Proceedings of the conference on Computational Linguistics (COLING'94)*, 1994.

Silberztein, M. Finite-state description of the French determiner system. *Journal of French Language Studies*, 13(2), 2003.

Tsuruoka, Y., J. Tsujii, and S. Ananiadou. Fast full parsing by linear-chain conditional random fields. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL'09)*, pages 790–798, 2009.

Woods, W.A. Transition network grammars for natural language analysis. *Communications of the ACM*, 13(10), 1970.

Address for correspondence:
Matthieu Constant
mconstan@univ-mlv.fr
Cité Descartes
5, boulevard Descartes
Champs-sur-Marne
77454 MARNE-LA-VALLEE Cedex 2
France