Multiword Expression Identification with Tree Substitution Grammars: A Parsing *tour de force* with French

Spence Green[*], **Marie-Catherine de Marneffe**[†], **John Bauer**[*], **and Christopher D. Manning**[*†]
[*]Computer Science Department, Stanford University
[†]Linguistics Department, Stanford University
{spenceg,mcdm,horatio,manning}@stanford.edu

Abstract

Multiword expressions (MWE), a known nuisance for both linguistics and NLP, blur the lines between syntax and semantics. Previous work on MWE identification has relied primarily on surface statistics, which perform poorly for longer MWEs and cannot model discontinuous expressions. To address these problems, we show that even the simplest parsing models can effectively identify MWEs of arbitrary length, and that Tree Substitution Grammars achieve the best results. Our experiments show a 36.4% F1 absolute improvement for French over an n-gram surface statistics baseline, currently the predominant method for MWE identification. Our models are useful for several NLP tasks in which MWE pre-grouping has improved accuracy.

1 Introduction

Multiword expressions (MWE) have long been a challenge for linguistic theory and NLP. There is no universally accepted definition of the term, but MWEs can be characterized as "idiosyncratic interpretations that cross word boundaries (or spaces)" (Sag et al., 2002) such as *traffic light*, or as "frequently occurring phrasal units which are subject to a certain level of semantic opaqueness, or non-compositionality" (Rayson et al., 2010).

MWEs are often opaque fixed expressions, although the degree to which they are fixed can vary. Some MWEs do not allow morphosyntactic variation or internal modification (e.g., *in short*, but **in shorter* or **in very short*). Other MWEs are "semi-fixed," meaning that they can be inflected or undergo internal modification. The type of modification is often limited, but not predictable, so it is not possible to enumerate all variants (Table 1).

French				English			
à			terme	in the		near	term
à		court	terme	in the		short	term
à	très	court	terme	in the	very	short	term
à		moyen	terme	in the		medium	term
à		long	terme	in the		long	term
à	très	long	terme	in the	very	long	term

Table 1: Semi-fixed MWEs in French and English. The French adverb *à terme* 'in the end' can be modified by a small set of adjectives, and in turn some of these adjectives can be modified by an adverb such as *très* 'very'. Similar restrictions appear in English.

Merging known MWEs into single tokens has been shown to improve accuracy for a variety of NLP tasks: dependency parsing (Nivre and Nilsson, 2004), constituency parsing (Arun and Keller, 2005), sentence generation (Hogan et al., 2007), and machine translation (Carpuat and Diab, 2010). Most experiments use gold MWE pre-grouping or language-specific resources like WordNet. For unlabeled text, the best MWE identification methods, which are based on surface statistics (Pecina, 2010), suffer from sparsity induced by longer n-grams (Ramisch et al., 2010). A dilemma thus exists: MWE knowledge is useful, but MWEs are hard to identify.

In this paper, we show the effectiveness of statistical parsers for MWE identification. Specifically, Tree Substitution Grammars (TSG) can achieve a 36.4% F1 absolute improvement over a state-of-the-art surface statistics method. We choose French, which has pervasive MWEs, for our experiments. Parsing models naturally accommodate discontinuous MWEs like phrasal verbs, and provide syntactic subcategorization. By contrast, surface statistics methods are usually limited to binary judgements for contiguous n-grams or dependency bigrams.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 725–735,
Edinburgh, Scotland, UK, July 27–31, 2011.

	FTB (train)	WSJ (train)
Sentences	13,449	39,832
Tokens	398,248	950,028
#Word Types	28,842	44,389
#Tag Types	30	45
#Phrasal Types	24	27
Per Sentence		
Depth (μ/σ^2)	4.03 / 0.360	4.18 / 0.730
Breadth (μ/σ^2)	13.5 / 6.79	10.7 / 4.59
Length (μ/σ^2)	29.6 / 17.3	23.9 / 11.2
Constituents (μ)	20.3	19.6
μ Constituents / μ Length	0.686	0.820

Table 2: Gross corpus statistics for the pre-processed FTB (training set) and WSJ (sec. 2-21). The FTB sentences are longer with broader syntactic trees. The FTB POS tag set has 33% fewer types than the WSJ. The FTB dev set OOV rate is 17.77% vs. 12.78% for the WSJ.

Type		#Total	#Single	%Single	%Total
MWN	*noun*	9,680	2,737	28.3	49.7
MWADV	*adverb*	3,852	449	11.7	19.8
MWP	*prep.*	3,526	342	9.70	18.1
MWC	*conj.*	814	73	8.97	4.18
MWV	*verb.*	585	243	41.5	3.01
MWD	*det.*	328	69	21.0	1.69
MWA	*adj.*	324	126	38.9	1.66
MWPRO	*pron.*	266	33	12.4	1.37
MWCL	*clitic*	59	1	1.69	0.30
MWET	*foreign*	24	18	0.75	0.12
MWI	*interj.*	4	2	0.50	0.02
		19,462	4,093	21.0%	100.0%

Table 3: Frequency distribution of the 11 MWE subcategories in the FTB (training set). MWEs account for 7.08% of the bracketings and 13.0% of the tokens in the treebank. Only 21% of the MWEs occur once ("single").

We first introduce a new instantiation of the French Treebank that, unlike previous work, does not use gold MWE pre-grouping. Consequently, our experimental results also provide a better baseline for parsing raw French text.

2 French Treebank Setup

The corpus used in our experiments is the French Treebank (Abeillé et al. (2003), version from June 2010, hereafter FTB). In French, there is a linguistic tradition of lexicography which compiles lists of MWEs occurring in the language. For example, Gross (1986) shows that dictionaries contain about 1,500 single-word adverbs but that French contains over 5,000 multiword adverbs. MWEs occur in every part-of-speech (POS) category (e.g., noun *trousse de secours* 'first-aid kit'; verb *faire main-basse* [do hand-low] 'seize'; adverb *comme dans du beurre* [as in butter] 'easily'; adjective *'à part entière'* 'wholly').

The FTB explicitly annotates MWEs (also called *compounds* in prior work). We used the subset of the corpus with functional annotations, not for those annotations but because this subset is known to be more consistently annotated. POS tags for MWEs are given not only at the MWE level, but also internally: most tokens that constitute an MWE also have a POS tag. Table 2 compares this part of the FTB to the WSJ portion of the Penn Treebank.

2.1 Preprocessing

The FTB requires significant pre-processing prior to parsing.

Tokenization We changed the default tokenization for numbers by fusing adjacent digit tokens. For example, *500 000* is tagged as an MWE composed of two words *500* and *000*. We made this *500000* and retained the MWE POS, although we did not mark the new token as an MWE. For consistency, we used one token for punctuated numbers like "17,9".

MWE Tagging We marked MWEs with a flat bracketing in which the phrasal label is the MWE-level POS tag with an "MW" prefix, and the preterminals are the internal POS tags for each terminal. The resulting POS sequences are not always unique to MWEs: they appear in abundance elsewhere in the corpus. However, some MWEs contain normally ungrammatical POS sequences (e.g., adverb *à la va vite* 'in a hurry': P D V ADV [at the goes quick]), and some words appear only as part of an MWE, such as *insu* in *à l'insu de* 'to the ignorance of'.

Labels We augmented the basic FTB label set—which contains 14 POS tags and 19 phrasal tags—in two ways. First, we added 16 finer-grained POS tags for punctuation.[1] Second, we added the 11 MWE

[1] Punctuation tag clusters—as used in the WSJ—did not improve accuracy. Enriched tag sets like that of Crabbé and Candito (2008) could also be investigated and compared to our results since Evalb is insensitive to POS tags.

labels shown in Table 3, resulting in 24 total phrasal categories.

Corrections Historically, the FTB suffered from annotation errors such as missing POS and phrasal tags (Arun and Keller, 2005). We found that this problem has been largely resolved in the current release. However, 1,949 tokens and 36 MWE spans still lacked tags. We restored the labels by first assigning each token its most frequent POS tag elsewhere in the treebank, and then assigning the most frequent MWE phrasal category for the resulting POS sequence.[2]

Split We used the 80/10/10 split described by Crabbé and Candito (2008). However, they used a previous release of the treebank with 12,531 trees. 3,391 trees have been added to the present version. We appended these extra trees to the training set, thus retaining the same development and test sets.

2.2 Comparison to Prior FTB Representations

Our pre-processing approach is simple and automatic[3] unlike the three major instantiations of the FTB that have been used in previous work:

Arun-Cont and Arun-Exp (Arun and Keller, 2005): Two instantiations of the full 20,000 sentence treebank that differed principally in their treatment of MWEs: (1) Cont, in which the tokens of each MWE were concatenated into a single token (*en moyenne* $\rightarrow$ *en_moyenne*); (2) Exp, in which they were marked with a flat structure. For both representations, they also gave results in which coordinated phrase structures were flattened. In the published experiments, they mistakenly removed half of the corpus, believing that the multi-terminal (per POS tag) annotations of MWEs were XML errors (Schluter and Genabith, 2007).

MFT (Schluter and Genabith, 2007): Manual revision to 3,800 sentences. Major changes included coordination raising, an expanded POS tag set, and the correction of annotation errors. Like Arun-Cont, MFT contains concatenated MWEs.

FTB-UC (Candito and Crabbé, 2009): An instantiation of the functionally annotated section that makes a distinction between MWEs that are "syntactically regular" and those that are not. Syntactically regular MWEs were given internal structure, while all other MWEs were concatenated into single tokens. For example, nouns followed by adjectives, such as *loi agraire* 'land law' or *Union monétaire et économique* 'monetary and economic Union' were considered syntactically regular. They are MWEs because the choice of adjective is arbitrary (*loi agraire* and not **loi agricole*, similarly to 'coal black' but not *'crow black' for example), but their syntactic structure is not intrinsic to MWEs. In such cases, FTB-UC gives the MWE a conventional analysis of an NP with internal structure. Such analysis is indeed sufficient to recover the meaning of these semantically compositional MWEs that are extremely productive. On the other hand, the FTB-UC loses information about MWEs with non-compositional semantics.

Almost all work on the FTB has followed Arun-Cont and used gold MWE pre-grouping. As a result, most results for French parsing are analogous to early results for Chinese, which used gold word segmentation, and Arabic, which used gold clitic segmentation. Candito et al. (2010) were the first to acknowledge and address this issue, but they still used FTB-UC (with some pre-grouped MWEs). Since the syntax and definition of MWEs is a contentious issue, we take a more agnostic view—which is consistent with that of the FTB annotators—and leave them tokenized. This permits a data-oriented approach to MWE identification that is more robust to changes to the status of specific MWE instances.

To set a baseline prior to grammar development, we trained the Stanford parser (Klein and Manning, 2003) with no grammar features, achieving 74.2% labeled F1 on the development set (sentences ≤ 40 words). This is lower than the most recent results obtained by Seddah (2010). However, the results are not comparable: the data split was different, they made use of morphological information, and more importantly they concatenated MWEs. The focus of

[2] 73 of the unlabeled word types did not appear elsewhere in the treebank. All but 11 of these were nouns. We manually assigned the correct tags, but we would not expect a negative effect by deterministically labeling all of them as nouns.

[3] We automate tree manipulation with Tregex/Tsurgeon (Levy and Andrew, 2006). Our pre-processing package is available at http://nlp.stanford.edu/software/lex-parser.shtml.

our work is on models and data representations that enable MWE identification.

3 MWEs in Lexicon-Grammar

The MWE representation in the FTB is close to the one proposed in the Lexicon-Grammar (Gross, 1986). In the Lexicon-Grammar, MWEs are classified according to their global POS tags (noun, verb, adverb, adjective), and described in terms of the sequence of the POS tags of the words that constitute the MWE (e.g., "N de N" *garde d'enfant* [guard of child] 'daycare', *pied de guerre* [foot of war] 'at the ready'). In other words, MWEs are represented by a flat structure. The Lexicon-Grammar distinguishes between units that are fixed and have to appear as is (*en tout et pour tout* [in all and for all] 'in total') and units that accept some syntactic variation such as admitting the insertion of an adverb or adjective, or the variation of one of the words in the expression (e.g., a possessive as in 'from the top of one's hat'). It also notes whether the MWE displays some selectional preferences (e.g., it has to be preceded by a verb or by an adjective).

Our FTB instantiation is largely consistent with the Lexicon-Grammar. Recall that we defined different MWE categories based on the global POS. We now detail three of the categories.

MWN The MWN category consists of proper nouns (1a), foreign common nouns (1b), as well as common nouns. The common nouns appear in several syntactically regular sequences of POS tags (2). Multiword nouns allow inflection (singular vs. plural) but no insertion.

(1) a. *London Sunday Times, Los Angeles*
b. *week - end, mea culpa, joint - venture*

(2) a. N A: *corps médical* 'medical staff', *dette publique* 'public debt'
b. N P N: *mode d'emploi* 'instruction manual'
c. N N: *numéro deux* 'number two', *maison mère* [house mother] 'headquarters', *grève surprise* 'sudden strike'
d. N P D N: *impôt sur le revenu* 'income tax', *ministre de l'économie* 'finance minister'

MWA Multiword adjectives appear with different POS sequences (3). They include numbers such as *vingt et unième* '21st'. Some items in (3b) allow internal variation: some adverbs or adjectives can be added to both examples given (*à <u>très</u> haut risque*, *de <u>toute</u> dernière minute*).

(3) a. P N: *d'antan* [from before] 'old', *en question* 'under discussion'
b. P A N: *à haut risque* 'high-risk', *de dernière minute* [from the last minute] 'at the eleventh hour'
c. A C A: *pur et simple* [pure and simple] 'straightforward', *noir et blanc* 'black and white'

MWV Multiword verbs also appear in several POS sequences (4). All verbs allow number and tense inflections. Some MWVs containing a noun or an adjective allow the insertion of a modifier (e.g., *donner grande satisfication* 'give great satisfaction'), whereas others do not. When an adverb intervenes between the main verb and its complement, the FTB marks the two parts of the MWV discontinuously (e.g., [MWV [V prennent]] [ADV déjà] [MWV [P en] [N cause]] 'already take into account').

(4) a. V N: *avoir lieu* 'take place', *donner satisfaction* 'give satisfaction'
b. V P N: *mettre en place* 'put in place', *entrer en vigueur* 'to come into effect'
c. V P ADV: *mettre à mal* [put at bad] 'harm', *être à même* [be at same] 'be able'
d. V D N P N: *tirer la sonnette d'alarme* 'ring the alarm bell', *avoir le vent en poupe* 'to have the wind astern'

4 Parsing Models

We develop two parsers for French with the goal of improving MWE identification. The first is a manually-annotated grammar that we incorporate into the Stanford parser. Manual annotation results in human interpretable grammars that can inform future treebank annotation decisions. Moreover, the grammar can be used as the base distribution in our second model, a Probabilistic Tree Substitution Grammar (PTSG) parser. PTSGs learn parameters for tree

Feature	States	Tags	F1	ΔF1
—	4325	31	74.21	
tagPA	4509	215	76.94	+2.73
markInf	4510	216	77.42	+0.48
markPart	4511	217	77.73	+0.31
markVN	5986	217	78.32	+0.59
markCoord	7361	217	78.45	+0.13
markDe	7521	233	79.11	+0.66
markP	7523	235	79.34	+0.23
markMWE	7867	235	79.23	−0.11

Table 4: Effects on grammar size and labeled F1 for each of the manual state splits (development set, sentences $\leq$ 40 words). *markMWE* decreases overall accuracy, but increases both the number of correctly parsed trees (by 0.30%) and per category MWE accuracy.

fragments larger than basic CFG rules. PTSG rules may also be lexicalized. This means that commonly observed collocations—some of which are MWEs—can be stored in the grammar.

4.1 Stanford Parser

We configure the Stanford parser with settings that are effective for other languages: selective parent annotation, lexicon smoothing, and factored parsing. We use the head-finding rules of Dybro-Johansen (2004), which we find to yield an approximately 1.0% F1 development set improvement over those of Arun (2004). Finally, we include a simple unknown word model consisting entirely of surface features:

- Nominal, adjectival, verbal, adverbial, and plural suffixes
- Contains a digit or punctuation
- Is capitalized (except the first word in a sentence)
- Consists entirely of capital letters
- If none of the above, add a one- or two-character suffix

Combined with the grammar features, this unknown word model yields 97.3% tagging accuracy on the development set.

4.1.1 Grammar Development

Table 4 lists the symbol refinements used in our grammar. Most of the features are POS splits as many phrasal tag splits did not lead to any improvement. Parent annotation of POS tags (*tagPA*) captures information about the external context. *markInf* and *markPart* accomplish a finite/nonfinite distinction: they respectively specify whether the verb is an infinitive or a participle based on the type of the grandparent node. *markVN* captures the notion of verbal distance as in Klein and Manning (2003).

We opted to keep the COORD phrasal tag, and to capture parallelism in coordination, we mark COORD with the type of its child (NP, AP, VPinf, etc.). *markDe* identifies the preposition *de* and its variants (*du*, *des*, *d'*) which is very frequent and appears in several different contexts. *markP* identifies prepositions which introduce PPs modifying a noun. Marking other kinds of prepositional modifiers (e.g., verb) did not help. *markMWE* adds an annotation to several MWE categories for frequently occuring POS sequences. For example, we mark MWNs that occur more than 600 times (e.g., "N P N" and "N N").

4.2 DP-TSG Parser

A shortcoming of CFG-based grammars is that they do not explicitly capture idiomatic usage. For example, consider the two utterances:

(5) a. He [$_{\text{MWV}}$ kicked the bucket] .
b. He [$_{\text{VP}}$ kicked [$_{\text{NP}}$ the pail]] .

The examples in (5) may be equally probable and receive the same analysis under a PCFG; words are generated independently. However, recall that in our representation, (5a) should receive a flat analysis as MWV, whereas (5b) should have a conventional analysis of the verb *kicked* and its two arguments.

An alternate view of parsing is one in which new utterances are built from previously observed fragments. This is the original motivation for data oriented parsing (DOP) (Bod, 1992), in which "idiomaticity is the rule rather than the exception" (Scha, 1990). If we have seen the collocation *kicked the bucket* several times before, we should store that whole fragment for later use.

We consider a variant of the non-parametric PTSG model of Cohn et al. (2009) in which tree fragments are drawn from a Dirichlet process (DP) prior.[4] The DP-TSG can be viewed as a DOP model with Bayesian parameter estimation. A PTSG is a 5-tuple $\langle V, \Sigma, R, \diamondsuit, \boldsymbol{\theta} \rangle$ where $c \in V$ are non-terminals;

[4]Similar models were developed independently by O'Donnell et al. (2009) and Post and Gildea (2009).

α_c	DP concentration parameter for each $c \in V$
$P_0(e\|c)$	CFG base distribution
$\boldsymbol{x}$	Set of non-terminal nodes in the treebank
$\mathcal{S}$	Set of sampling sites (one for each $x \in \boldsymbol{x}$)
S	A block of sampling sites, where $S \subseteq \mathcal{S}$
$\boldsymbol{b} = \{b_s\}_{s \in \mathcal{S}}$	Binary variables to be sampled ($b_s = 1 \rightarrow$ frontier node)
$\boldsymbol{z}$	Latent state of the segmented treebank
m	Number of sites $s \in S$ s.t. $b_S = 1$
$\boldsymbol{n} = \{n_{c,e}\}$	Sufficient statistics of $\boldsymbol{z}$
$\Delta n^{S:m}$	Change in counts by setting m sites in S

Table 5: DP-TSG model notation. For consistency, we largely follow the notation of Liang et al. (2010). Note that $\boldsymbol{z} = (\boldsymbol{b}, \boldsymbol{x})$, and as such $z = \langle c, e \rangle$.

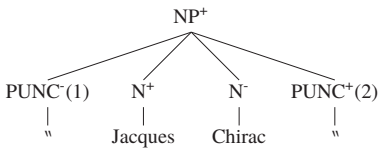


Figure 1: Example of two conflicting sites of the same type. Define the *type* of a site $t(\boldsymbol{z}, s) \stackrel{\text{def}}{=} (\Delta n^{s:0}, \Delta n^{s:1})$. Sites (1) and (2) above have the same type since $t(\boldsymbol{z}, s_1) = t(\boldsymbol{z}, s_2)$. However, the two sites *conflict* since the probabilities of setting b_{s_1} and b_{s_2} both depend on counts for the tree fragment rooted at NP. Consequently, sites (1) and (2) are not exchangeable: the probabilities of their assignments depend on the order in which they are sampled.

$t \in \Sigma$ are terminals; $e \in R$ are elementary trees;[5] $\diamondsuit \in V$ is a unique start symbol; and $\theta_{c,e} \in \boldsymbol{\theta}$ are parameters for each tree fragment. A PTSG derivation is created by successively applying the substitution operator to the leftmost *frontier node* (denoted by c^+). All other nodes are *internal* (denoted by c^-).

In the supervised setting, DP-TSG grammar extraction reduces to a segmentation problem. We have a treebank T that we segment into the set R, a process that we model with Bayes' rule:

$$p(R \mid T) \propto p(T \mid R)\, p(R) \tag{1}$$

Since the tree fragments completely specify each tree, $p(T \mid R)$ is either 0 or 1, so all work is performed by the prior over the set of elementary trees.

The DP-TSG contains a DP prior for each $c \in V$ (Table 5 defines further notation). We generate $\langle c, e \rangle$ tuples as follows:

$$\theta_c | c, \alpha_c, P_0(\cdot|c) \sim DP(\alpha_c, P_0)$$
$$e | \theta_c \sim \theta_c$$

The data likelihood is given by the latent state $\boldsymbol{z}$ and the parameters $\boldsymbol{\theta}$: $p(\boldsymbol{z}|\boldsymbol{\theta}) = \prod_{z \in \boldsymbol{z}} \theta_{c,e}^{n_{c,e}(\boldsymbol{z})}$. Integrating out the parameters, we have:

$$p(\boldsymbol{z}) = \prod_{c \in V} \frac{\prod_e (\alpha_c P_0(e|c))^{\overline{n_{c,e}(\boldsymbol{z})}}}{\alpha_c^{\overline{n_{c,\cdot}(\boldsymbol{z})}}} \tag{2}$$

where $x^{\overline{n}} = x(x+1)\ldots(x+n-1)$ is the rising factorial. (§A.1 contains ancillary details.)

Base Distribution The base distribution P_0 is the same maximum likelihood PCFG used in the Stanford parser.[6,7] After applying the manual state splits, we perform simple right binarization, collapse unary rules, and replace rare words with their signatures (Petrov et al., 2006).

For each non-terminal type c, we learn a stop probability $s_c \sim \text{Beta}(1, 1)$. Under P_0, the probability of generating a rule $A^+ \rightarrow B^-\ C^+$ composed of non-terminals is

$$P_0(A^+ \rightarrow B^-\ C^+) = p_{\text{MLE}}(A \rightarrow B\ C) s_B (1 - s_C) \tag{3}$$

For lexical insertion rules, we add a penalty proportional to the frequency of the lexical item:

$$P_0(c \rightarrow t) = p_{\text{MLE}}(c \rightarrow t) p(t) \tag{4}$$

where $p(t)$ is equal to the MLE unigram probability of t in the treebank. Lexicalizing a rule makes it very specific, so we generally want to avoid lexicalization with rare words. Empirically, we found that this penalty reduces overfitting.

Type-based Inference Algorithm To learn the parameters $\boldsymbol{\theta}$ we use the collapsed, block Gibbs sampler of Liang et al. (2010). We sample binary variables b_s associated with each non-terminal node/site in the treebank. The key idea is to select a block of exchangeable sites S of the same *type* that do not *conflict* (Figure 1). Since the sites in S are exchangeable, we can set $\boldsymbol{b}_S$ randomly so long as we know m, the number of sites with $b_s = 1$. Because this algorithm is a not a contribution of this paper, we refer the reader to Liang et al. (2010).

[5] We use the terms *tree fragment* and *elementary tree* interchangeably.

[6] The Stanford parser is a product model, so the results in §5.1 include the contribution of a dependency parser.

[7] Bansal and Klein (2010) also experimented with symbol refinement in an all-fragments (parametric) TSG for English.

After each Gibbs iteration, we sample each s_c directly using binomial-Beta conjugacy. We re-sample the DP concentration parameters α_c with the auxiliary variable procedure of West (1995).

Decoding We compute the rule score of each tree fragment from a single grammar sample as follows:

$$\theta_{c,e} = \frac{n_{c,e}(\boldsymbol{z}) + \alpha_c P_0(e|c)}{n_{c,\cdot}(\boldsymbol{z}) + \alpha_c} \quad (5)$$

To make the grammar more robust, we also include all CFG rules in P_0 with zero counts in $\boldsymbol{n}$. Scores for these rules follow from (5) with $n_{c,e}(\boldsymbol{z}) = 0$.

For decoding, we note that the derivations of a TSG are a CFG parse forest (Vijay-Shanker and Weir, 1993). As such, we can use a Synchronous Context Free Grammar (SCFG) to translate the 1-best parse to its derivation. Consider a unique tree fragment e_i rooted at X with frontier γ, which is a sequence of terminals and non-terminals. We encode this fragment as an SCFG rule of the form

$$[X \rightarrow \gamma \ , \ X \rightarrow i, Y_1, \ldots, Y_n] \quad (6)$$

where $Y_1, \ldots, Y_n$ is the sequence of non-terminal nodes in γ.[8] During decoding, the input is rewritten as a sequence of tree fragment (rule) indices $\{i, j, k, \ldots\}$. Because the TSG substitution operator always applies to the leftmost frontier node, we can deterministically recover the monolingual parse with top-down re-writes of $\diamondsuit$.

The SCFG formulation has a practical benefit: we can take advantage of the heavily-optimized SCFG decoders for machine translation. We use `cdec` (Dyer et al., 2010) to recover the Viterbi derivation under a DP-TSG grammar sample.

5 Experiments

5.1 Standard Parsing Experiments

We evaluate parsing accuracy of the Stanford and DP-TSG models (Table 6). For comparison, we also include the Berkeley parser (Petrov et al., 2006).[9] For the DP-TSG, we initialized all b_s with fair coin tosses and ran for 400 iterations, after which likelihood stopped improving.

[8] This formulation is due to Chris Dyer.

[9] Training settings: right binarization, no parent annotation, six split-merge cycles, and random initialization.

	Leaf Ancestor		Evalb			
	Corpus	Sent	LP	LR	F1	EX%
PA-PCFG	0.793	0.812	68.1	67.0	67.6	10.5
DP-TSG	0.823	0.842	75.6	76.0	75.8	15.1
Stanford	0.843	0.861	77.8	79.0	78.4	17.5
Berkeley	**0.880**	**0.891**	**82.4**	**82.0**	**82.2**	**21.4**

Table 6: Standard parsing experiments (test set, sentences $\leq$ 40 words). All parsers exceed 96% tagging accuracy. Berkeley and DP-TSG results are the average of three independent runs.

We report two different parsing metrics. *Evalb* is the standard labeled precision/recall metric.[10] *Leaf Ancestor* measures the cost of transforming guess trees to the reference (Sampson and Babarczy, 2003). It was developed in response to the non-terminal/terminal ratio bias of Evalb, which penalizes flat treebanks like the FTB. The range of the score is between 0 and 1 (higher is better). We report micro-averaged (whole corpus) and macro-averaged (per sentence) scores.

In terms of parsing accuracy, the Berkeley parser exceeds both Stanford and DP-TSG. This is consistent with previous experiments for French by Seddah et al. (2009), who show that the Berkeley parser outperforms other models. It also matches the ordering for English (Cohn et al., 2010; Liang et al., 2010). However, the standard baseline for TSG models is a simple parent-annotated PCFG (PA-PCFG). For English, Liang et al. (2010) showed that a similar DP-TSG improved over PA-PCFG by 4.2% F1. For French, our gain is a more substantial 8.2% F1.

5.2 MWE Identification Experiments

Table 7 lists overall and per-category MWE identification results for the parsing models. Although DP-TSG is less accurate as a general parsing model, it is more effective at identifying MWEs.

The predominant approach to MWE identification is the combination of lexical association measures (surface statistics) with a binary classifier (Pecina, 2010). A state-of-the-art, language independent package that implements this approach for higher order n-grams is `mwetoolkit` (Ramisch et al., 2010).[11] In Table 8 we compare DP-TSG to both

[10] Available at http://nlp.cs.nyu.edu/evalb/ (v.20080701).

[11] Available at http://multiword.sourceforge.net/. See §A.2 for

	#gold	Stanford	DP-TSG	Berkeley
MWET	3	0.0	0.0	0.0
MWV	26	64.0	57.7	50.7
MWA	8	26.1	32.2	29.8
MWN	456	64.1	67.6	67.1
MWD	15	70.3	65.5	70.1
MWPRO	17	73.7	78.0	76.2
MWADV	220	74.6	72.7	70.4
MWP	162	81.3	80.5	77.7
MWC	47	83.5	83.5	80.8
	954	70.1	**71.1**	69.6

Table 7: MWE identification per category and overall results (test set, sentences ≤ 40 words). MWI and MWCL do not occur in the test set.

Model	F1
`mwetoolkit` All	15.4
PA-PCFG	32.6
`mwetoolkit` Filter	34.7
PA-PCFG+Features	63.1
DP-TSG	**71.1**

Table 8: MWE identification F1 of the best parsing model vs. the `mwetoolkit` baseline (test set, sentences ≤ 40 words). PA-PCFG+Features includes the grammar features in Table 4, which is the CFG from which the TSG is extracted. For `mwetoolkit`, *All* indicates the inclusion of all n-grams in the training corpus. *Filter* indicates prefiltering of the training corpus by removing rare n-grams (see §A.2 for details).

`mwetoolkit` and the CFG from the which the TSG is extracted. The TSG-based parsing model outperforms `mwetoolkit` by 36.4% F1 while providing syntactic subcategory information.

6 Discussion

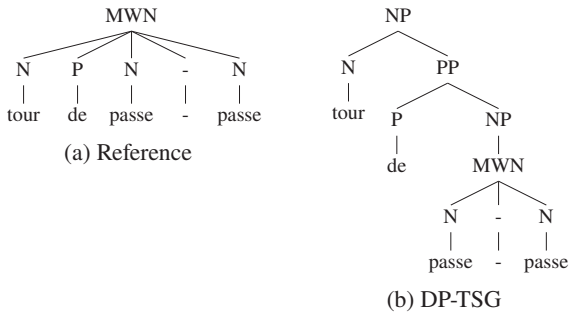
Automatic learning methods run the risk of producing uninterpretable models. However, the DP-TSG model learns useful generalizations over MWEs. A sample of the rules is given in Table 9. Some specific sequences like "[MWN [coup de N]]" are part of the grammar: such rules can indeed generate quite a few MWEs, e.g., *coup de pied* 'kick', *coup de coeur*, *coup de foudre* 'love at first sight', *coup de main* 'help', *coup d'état*, *coup de grâce* (note that only some of these MWEs are seen in the training data). For MWV, "V de N" as in *avoir de cesse* 'give no peace', *perdre de vue* [lose from sight] 'forget', *prendre de vitesse* [take from speed] 'outpace'), is learned. For prepositions, the grammar stores full subtrees of MWPs, but can also generalize the structure of very frequent sequences: "en N de" occurs in many multiword prepositions (e.g., *en compagnie de*, *en face de*, *en matière de*, *en terme de*, *en cours de*, *en faveur de*, *en raison de*, *en fonction de*). The TSG grammar thus provides a categorization of MWEs consistent with the Lexicon-Grammar. It also learns verbal phrases which contain discontinuous MWVs due to the insertion of an adverb or negation such as "[VN [MWV va] [MWADV d'ailleurs] [MWV bon train]]" [go indeed well], "[VN [MWV a] [ADV jamais] [MWV été question d']]" [has never been in question].

MWN
sociétés de N
prix de N
coup de N
N d'état
N de N
N à N

MWV
sous - V
faire N
V les moyens
V de N
V en N

MWP
de l'ordre de
y compris
au N de
en N de
ADV de

Table 9: Sample of the TSG rules learned.



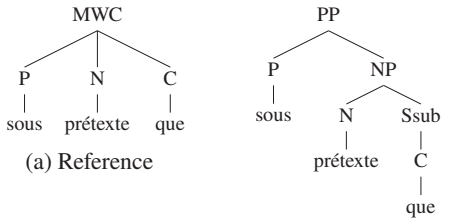
(a) Reference (b) DP-TSG

Figure 2: Example of an MWE error for *tour de passe-passe* 'magic trick'. (dev set)

A significant fraction of errors for MWNs occur with adjectives that are not recognized as part of the MWE. For example, since *établissements privés* 'private corporation' is unseen in the training data, it is not found. Sometimes the parser did not recognize the whole structure of an MWE. Figure 2 shows an example where the parser only found a subpart of the MWN *tour de passe-passe* 'magic trick'.

Other DP-TSG errors are due to inconsistencies in the FTB annotation. For example, *sous prétexte que*

configuration details.



(a) Reference

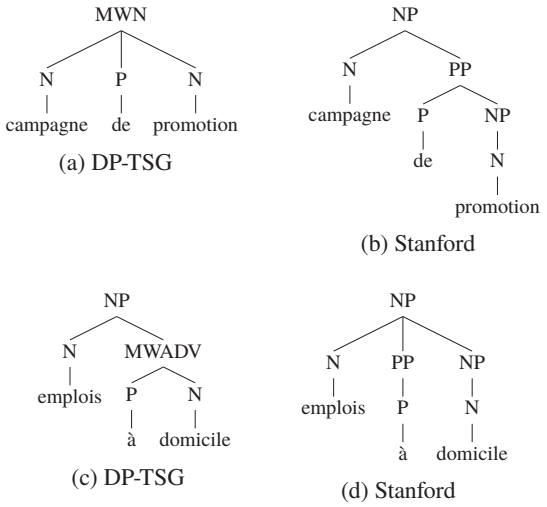
(b) Reference

Figure 3: Example of an inconsistent FTB annotation for *sous prétexte que* 'on the pretext of'.

'on the pretext of' is tagged as both MWC and as a regular PP structure (Figure 3). However, the parser always assigns a MWC structure, which is a better analysis than the gold annotation. We expect that more consistent annotation would help the DP-TSG more than the CFG-based parsers.

The DP-TSG is not immune to false positives: in *Le marché national, fait-on remarquer, est enfin en régression* ... 'The national economy, people at last note, is going down' the parser tags *marché national* as MWN. As noted, the boundary of what should and should not count as an MWE can be fuzzy, and it is therefore hard to assess whether or not this should be an MWE. The FTB does not mark it as one.

There are multiple examples were the DP-TSG found the MWE whereas Stanford (its base distribution) did not, such as in Figure 4. Note that the "N P N" structure is quite frequent for MWNs, but the TSG correctly identifies the MWADV in *emplois à domicile* [jobs at home] 'homeworking'.



(a) DP-TSG

(b) Stanford

(c) DP-TSG

(d) Stanford

Figure 4: Correct analyses by DP-TSG. (dev set)

7 Related Work

There is a voluminous literature on MWE identification. Here we review closely related syntax-based methods.[12] The linguistic and computational attractiveness of lexicalized grammars for modeling idiosyncratic constructions in French was identified by Abeillé (1988) and Abeillé and Schabes (1989). They manually developed a small Tree Adjoining Grammar (TAG) of 1,200 elementary trees and 4,000 lexical items that included MWEs. The classic statistical approach to MWE identification, Xtract (Smadja, 1993), used an incremental parser in the third stage of its pipeline to identify predicate-argument relationships. Lin (1999) applied information-theoretic measures to automatically-extracted dependency relations to find MWEs. To our knowledge, Wehrli (2000) was the first to use syntactically annotated corpora to improve a parser for MWE identification. He proposed to rank analyses of a symbolic parser based on the presence of collocations, although details of the ranking function were not provided.

The most similar work to ours is that of Nivre and Nilsson (2004), who converted a Swedish corpus into two versions: one in which MWEs were left as tokens, and one in which they were merged. On the first version, they showed that a deterministic dependency parser could identify MWEs at 71.1% F1, albeit without subcategory information. On the second version—which simulated perfect MWE identification—they showed that labeled attachment improved by about 1%.

Recent statistical parsing work on French has included Stochastic Tree Insertion Grammars (STIGs), which are related to TAGs, but with a restricted adjunction operation.[13] Seddah et al. (2009) and Seddah (2010) showed that STIGs underperform CFG-based parsers on the FTB. In their experiments, MWEs were concatenated.

[12] See Seretan (2011) for a comprehensive survey of syntax-based methods for MWE identification. For an overview of n-gram methods like `mwetoolkit`, see Pecina (2010).

[13] TSGs differ from TAGs and STIGs in that they do not include an adjunction operator.

8 Conclusion

The main result of this paper is that an existing statistical parser can achieve a 36.4% F1 absolute improvement for MWE identification over a state-of-the-art n-gram surface statistics package. Parsers also provide syntactic subcategorization, and do not require pre-filtering of the training data. We have also demonstrated that TSGs can capture idiomatic usage better than a PCFG. While the DP-TSG, which is a relatively new parsing model, still lags state-of-the-art parsers in terms of overall labeling accuracy, we have shown that it is already very effective for other tasks like MWE identification. We plan to improve the DP-TSG by experimenting with alternate parsing objectives (Cohn et al., 2010), lexical representations, and parameterizations of the base distribution. A particularly promising base distribution is the latent variable PCFG learned by the Berkeley parser. However, initial experiments with this distribution were negative, so we leave further development to future work.

We chose French for these experiments due to the pervasiveness of MWEs and the availability of an annotated corpus. However, MWE lists and syntactic treebanks exist for many of the world's major languages. We will investigate automatic conversion of these treebanks (by flattening MWE bracketings) for MWE identification.

A Appendix

A.1 Notes on the Rising Factorial

The rising factorial—also known as the ascending factorial or Pochhammer symbol—arises in the context of samples from a Dirichlet process (see Prop. 3 of Antoniak (1974) for details). For a positive integer n and a complex number x, the rising factorial $x^{\overline{n}}$ is defined[14] by

$$\begin{aligned} x^{\overline{n}} &= x(x+1)\dots(x+n-1) \\ &= \prod_{j=1}^{n}(x+j-1) \end{aligned} \tag{7}$$

The rising factorial can be generalized to a complex number α with the gamma function:

$$x^{\overline{\alpha}} = \frac{\Gamma(x+\alpha)}{\Gamma(x)} \tag{8}$$

where $x^{\overline{0}} \equiv 1$.

In our type-based sampler, we computed (7) directly in a dynamic program. We found that (8) was prohibitively slow for sampling.

A.2 `mwetoolkit` Configuration

We configured `mwetoolkit`[15] with the four standard lexical features: the maximum likelihood estimator, Dice's coefficient, pointwise mutual information (PMI), and Student's t-score. We added the POS sequence for each n-gram as a single feature. We removed the web counts features to make the experiments comparable. To compensate for the absence of web counts, we computed the lexical features using the gold lemmas from the FTB instead of using an automatic lemmatizer.

Since MWE n-grams only account for a small fraction of the n-grams in the corpus, we filtered the training and test sets by removing all n-grams that occurred once. To further balance the proportion of MWEs, we trained on all valid MWEs plus 10x randomly selected non-MWE n-grams. This proportion matches the fraction of MWE/non-MWE tokens in the FTB. Since we generated a random training set, we reported the average of three independent runs.

We created feature vectors for the training n-grams and trained a binary Support Vector Machine (SVM) classifier with Weka (Hall et al., 2009). Although `mwetoolkit` defaults to a linear kernel, we achieved higher accuracy on the development set with an RBF kernel.

The FTB is sufficiently large for the corpus-based methods implemented in `mwetoolkit`. Ramisch et al. (2010)'s experiments were on Genia, which contains 18k sentences and 490k tokens, similar to the FTB. Their test set had 895 sentences, smaller than ours. They reported 30.6% F1 for their task against an Xtract baseline, which only obtained 7.3% F1. These results are comparable in magnitude to our FTB results.

Acknowledgments We thank Marie Candito, Chris Dyer, Dan Flickinger, Percy Liang, Carlos Ramisch, Djamé Seddah, and Val Spitkovsky for their helpful comments. The first author is supported by a National Defense Science and Engineering Graduate (NDSEG) fellowship.

[14] We adopt the notation of Knuth (1992).

[15] We re-implemented `mwetoolkit` in Java for compatibility with Weka and our pre-processing routines.

References

A. Abeillé and Y. Schabes. 1989. Parsing idioms in lexicalized TAGs. In *EACL*.

A. Abeillé, L. Clément, and A. Kinyon, 2003. *Building a treebank for French*, chapter 10. Kluwer.

A. Abeillé. 1988. Parsing French with Tree Adjoining Grammar: some linguistic accounts. In *COLING*.

C. E. Antoniak. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174.

A. Arun and F. Keller. 2005. Lexicalization in crosslinguistic probabilistic parsing: The case of French. In *ACL*.

A. Arun. 2004. Statistical parsing of the French treebank. Technical report, University of Edinburgh.

M. Bansal and D. Klein. 2010. Simple, accurate parsing with an all-fragments grammar. In *ACL*.

R. Bod. 1992. A computation model of language performance: Data-Oriented Parsing. In *COLING*.

M. Candito and B. Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *IWPT*.

M. Candito, B. Crabbé, and P. Denis. 2010. Statistical French dependency parsing: treebank conversion and first results. In *LREC*.

M. Carpuat and M. Diab. 2010. Task-based evaluation of multiword expressions: a pilot study in statistical machine translation. In *HLT-NAACL*.

T. Cohn, S. Goldwater, and P. Blunsom. 2009. Inducing compact but accurate tree-substitution grammars. In *HLT-NAACL*.

T. Cohn, P. Blunsom, and S. Goldwater. 2010. Inducing tree-substitution grammars. *JMLR*, 11:3053–3096, Nov.

B. Crabbé and M. Candito. 2008. Expériences d'analyse syntaxique statistique du français. In *TALN*.

A. Dybro-Johansen. 2004. Extraction automatique de grammaires à partir d'un corpus français. Master's thesis, Université Paris 7.

C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, et al. 2010. `cdec`: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *ACL System Demonstrations*.

M. Gross. 1986. Lexicon-Grammar: the representation of compound words. In *COLING*.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11:10–18.

D. Hogan, C. Cafferkey, A. Cahill, and J. van Genabith. 2007. Exploiting multi-word units in history-based probabilistic generation. In *EMNLP-CoNLL*.

D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. In *ACL*.

D. E. Knuth. 1992. Two notes on notation. *American Mathematical Monthly*, 99:403–422, May.

R. Levy and G. Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *LREC*.

P. Liang, M. I. Jordan, and D. Klein. 2010. Type-based MCMC. In *HLT-NAACL*.

D. Lin. 1999. Automatic identification of non-compositional phrases. In *ACL*.

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

T. J. O'Donnell, J. B. Tenenbaum, and N. D. Goodman. 2009. Fragment grammars: Exploring computation and reuse in language. Technical report, MIT Computer Science and Artificial Intelligence Laboratory Technical Report Series, MIT-CSAIL-TR-2009-013.

P. Pecina. 2010. Lexical association measures and collocation extraction. *Language Resources and Evaluation*, 44:137–158.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *ACL*.

M. Post and D. Gildea. 2009. Bayesian learning of a tree substitution grammar. In *ACL-IJCNLP, Short Papers*.

C. Ramisch, A. Villavicencio, and C. Boitet. 2010. `mwetoolkit`: a framework for multiword expression identification. In *LREC*.

P. Rayson, S. Piao, S. Sharoff, S. Evert, and B. Moirón. 2010. Multiword expressions: hard going or plain sailing? *Language Resources and Evaluation*, 44:1–5.

I. A. Sag, T. Baldwin, F. Bond, A. Copestake, and D. Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *CICLing*.

G. Sampson and A. Babarczy. 2003. A test of the leaf-ancestor metric for parse accuracy. *Natural Language Engineering*, 9:365–380.

R. Scha, 1990. *Taaltheorie en taaltechnologie: competence en performance*, pages 7–22. Landelijke Vereniging van Neerlandici (LVVNjaarboek).

N. Schluter and J. Genabith. 2007. Preparing, restructuring, and augmenting a French treebank: Lexicalised parsers or coherent treebanks? In *Pacling*.

D. Seddah, M. Candito, and B. Crabbé. 2009. Cross parser evaluation and tagset variation: a French treebank study. In *IWPT*.

D. Seddah. 2010. Exploring the Spinal-STIG model for parsing French. In *LREC*.

V. Seretan. 2011. *Syntax-Based Collocation Extraction*, volume 44 of *Text, Speech, and Language Technology*. Springer.

F. Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19:143–177.

K. Vijay-Shanker and D. J. Weir. 1993. The use of shared forests in tree adjoining grammar parsing. In *EACL*.

E. Wehrli. 2000. Parsing and collocations. In *Natural Language Processing—NLP 2000*, volume 1835 of *Lecture Notes in Computer Science*, pages 272–282. Springer.

M. West. 1995. Hyperparameter estimation in Dirichlet process mixture models. Technical report, Duke University.