Verbal Multiword Expression Identification: Do We Need a Sledgehammer to Crack a Nut?

Caroline Pasquer, Agata Savary, Carlos Ramisch, Jean-Yves Antoine

Verbal Multiword Expression Identification: Do We Need a Sledgehammer to Crack a Nut?

Caroline Pasquer
University of Tours, LIFAT
France
first.last@etu.univ-tours.fr

Agata Savary
University of Tours, LIFAT
France
first.last@univ-tours.fr

Carlos Ramisch
Aix Marseille Univ, Université de Toulon,
CNRS, LIS, Marseille, France
first.last@lis-lab.fr

Jean-Yves Antoine
University of Tours, LIFAT
France
first.last@univ-tours.fr

Abstract

Automatic identification of multiword expressions (MWEs), like *to* ***cut corners*** '*to do an incomplete job*', is a pre-requisite for semantically-oriented downstream applications. This task is challenging because MWEs, especially verbal ones (VMWEs), exhibit surface variability. This paper deals with a subproblem of VMWE identification: the identification of occurrences of previously seen VMWEs. A simple language-independent system based on a combination of filters competes with the best systems from a recent shared task: it obtains the best averaged F-score over 11 languages (0.6653) and even the best score for both seen and unseen VMWEs due to the high proportion of seen VMWEs in texts. This highlights the fact that focusing on the identification of seen VMWEs could be a strategy to improve VMWE identification in general.

1 Introduction

Multiword expressions (MWEs) are word combinations idiosyncratic with respect to e.g. syntax or semantics (Baldwin and Kim, 2010). One of their most emblematic properties is semantic non-compositionality: the meaning of the whole cannot be straightforwardly deduced from the meanings of its components, as in ***cut corners*** '*do an incomplete job*'.[1] Due to this property and to their frequency (Jackendoff, 1997), MWEs are a major challenge for semantically-oriented downstream applications, such as machine translation. A prerequisite for an MWE processing is their automatic identification.

MWE identification aims at locating MWE occurrences in running text. This task is very challenging, as signaled by Constant et al. (2017), and further confirmed by the PARSEME shared task on automatic identification of verbal MWEs (Ramisch et al., 2018). One of the main difficulties stems from the variability of MWEs, especially verbal ones (VMWEs). That is, even if a VMWE has previously been observed in a training corpus or in a lexicon, it can re-appear in morphosyntactically diverse forms. Examples (1–2) show two occurrences of a VMWE with variation in the components' inflection (*cutting* vs. *cut*), word order, presence of discontinuities (*were*), and syntactic relations (obj vs. nsubj).

(1) Some companies were **cutting corners**$_{\text{obj}}$ to save costs.
(2) The field would look uneven if **corners**$_{\text{nsubj}}$ were **cut**.

However, unrestricted variability is not a reasonable assumption either, since it may lead to literal or coincidental occurrences of VMWEs' components (Savary et al., 2019b), as in (3) and (4), respectively.[2]

(3) Start with cutting one corner of the disinfectant bag.
(4) If you cut along these lines, you'll get two acute corners.

[1] Henceforth, the lexicalized components of a MWE, i.e. those always realized by the same lexemes, appear in bold.
[2] Henceforth, literal and coincidental occurrences are highlighted with wavy underlining, following Savary et al. (2019b).

	train		dev				test			
	# tokens	# VMWEs	# tokens	# VMWEs	# seen	% seen	# tokens	# VMWEs	# seen	% seen
FR	432389	4550	56254	629	485	77.1	39489	498	251	50.4
PL	220465	4122	26030	515	387	75.1	27823	515	371	72.0
PT	506773	4430	68581	553	409	74.0	62648	553	397	71.8
RO	781968	4713	118658	589	555	94.2	114997	589	561	92.2

Table 1: PARSEME shared task corpora for the 4 languages in focus (FR, PL, PT, RO) in terms of the number of tokens, annotated VMWEs and seen VMWEs (those whose multiset of lemmas also appear annotated in `train`).

Our paper addresses VMWE variability, so as to distinguish examples (1-2) from (3-4). We focus on a subproblem of VMWE identification: the identification of previously seen VMWEs. Section 2 describes the corpora and best systems of the PARSEME shared task 1.1, Sections 3 and 4 motivate and describe our system *Seen*2020 dedicated to the task of seen VMWE identification. Experimental results are shown in Section 5, an interpretation is proposed in Section 6 and we conclude in Section 7.

2 PARSEME Shared Task 2018

VMWE identification recently received attention, especially in the PARSEME community, with the development and release of multilingual annotation guidelines and annotated corpora.[3] These data underlie three editions of the PARSEME shared task, dedicated to VMWE identification. This section discusses the second edition (1.1); the third edition (1.2) was ongoing at the time of writing.

Corpora The PARSEME corpora contain surface forms, lemmas, parts of speech (POS), morphological features, syntactic dependencies and VMWE annotations.[4] VMWEs are categorized into verbal idioms (VID: ***cut corners***), light-verb constructions (LVC.full: *to* ***take*** *a* ***walk***), inherently reflexive verbs (IRV: ***s'apercevoir*** '*to perceive oneself*'⇒'to realize' in French), etc. The categories cover various syntactic patterns, not only VERB-NOUN pairs as in some related work (Fazly et al., 2009). The corpora contain mostly newspaper texts, but differences in category distributions are due to domain, size, topic, and language structures (Savary et al., 2018). Corpora are split into training sets (`train`), development sets (`dev`, unavailable for some languages) and `test` sets.

Among the 19 languages of the PARSEME shared task 1.1, we focus on 10 which: (i) benefit from a `dev` corpus, (ii) do not suffer from lacking lemmas and (iii) are also covered in edition 1.2:[5] Bulgarian (BG), Basque (EU), French (FR), Polish (PL), Brazilian Portuguese (PT), Romanian (RO), German (DE), Greek (EL), Hebrew (HE) and Italian (IT). MWE components are not always separated by spaces or hyphens. Such single-token VMWEs are particularly frequent in Hungarian (HU), corresponding to 74% of the VMWEs (***határozathozatal*** 'take decision'). In order to study this phenomenon, Hungarian was added to the selected languages. Our experiments use the corpora from edition 1.1, so as to compare our results with state-of-the-art systems (results of edition 1.2 were not available at the time of writing).

Task Edition 1.1 of the PARSEME shared task aimed at boosting the development of VMWE identifiers (Ramisch et al., 2018). Depending on the use of external resources, systems participated in the *open* or *closed* tracks. In the closed track, systems could only use the annotated corpora described above.

The task has three phases. In the training phase, participants are given, for all languages, `train` and `dev` corpora annotated for VMWEs. In the prediction phase, participants are given `test` corpora in blind mode, that is, with all annotations except VMWEs, which must then be predicted. In the evaluation phase, predictions are compared to the manually annotated reference in `test`. Morphosyntactic annotations (e.g. lemmas, POS) in `test` are available to participants in the prediction phase, but their quality varies across languages, since for some languages they are generated automatically.

In this framework, the subtask of seen VMWE identificaiton focuses on *seen VMWEs*. A VMWE in `dev` or `test` is considered seen if a VMWE with the same multiset of lemmas is annotated at least once in `train`. Multisets are preferred over simple sets to represent repeated lemmas, e.g. ***hand*** *in* ***hand***

[3] https://gitlab.com/parseme/corpora/-/wikis

[4] VMWE annotation is manual, morphosyntax may be automatic: http://hdl.handle.net/11372/LRT-2842

[5] http://multiword.sourceforge.net/sharedtask2020/

(Ramisch et al., 2018).[6] Table 1 shows statistics for corpora of 4 languages on which we focus. Seen VMWEs represent 74% to 94% of VMWEs in `dev` and 50% to 92% in `test`. Due to this prevalence, improvements in seen VMWE identification can benefit VMWE identification in general.

Best Systems We compare our system to four PARSEME shared task 1.1 systems selected for their high performances (for at least one language) and a variety of architectures. The systems belonging to the closed track are $TRAVERSAL$ (Waszczuk, 2018), $TRAPACC_S$ (Stodden et al., 2018) and $VarIDE$ (Pasquer et al., 2018a), whereas $SHOMA$ (Taslimipoor and Rohanian, 2018) competed in the open track because it uses pre-trained word embeddings. $TRAVERSAL$ searches the optimal labeling of syntactic trees via multiclass logistic regression. In $VarIDE$, a naive Bayes classifier uses morphosyntactic information to predict if extracted candidates could be idiomatic. $TRAPACC_S$ and $SHOMA$ rely on a neural architecture: convolutional layers and SVM for the former, convolutional and recurrent layers with an optional CRF layer for the latter. Rohanian et al. (2019) propose an improvement of $SHOMA$, hereafter $SHOMA$ 2019, conceived to better handle discontinuities. This optimised version relies on two neural architectures (graph convolutional network and multi-head self-attention) combined or applied separately, and focuses on 4 languages: DE, EN, FA and FR.

3 MWEs' Nature as a Guiding Principle

MWE identification is a hard task, as exemplified by the PARSEME shared task results, in which the best systems, (Sec. 2) achieve global cross-language macro-averaged F1 scores below 0.6. Savary et al. (2019a) argue that this is mainly due to the very nature of MWEs. Namely, MWEs of the general language (as opposed to specialized phenomena such as named entities and multiword terms) are mostly regular at the level of tokens (individual occurrences), but idiosyncratic at the level of types (sets of occurrences).[7] For instance, the MWEs in (1)–(2) are perfectly regular English constructions, containing no special capitalization or trigger words, but their comparison to (3)–(4) reveals the prohibited number inflection of the noun *corners*. Additionally to this *type-level idiosyncrasy*, general-language MWEs are often dissimilar among each other but similar to regular (non-MWE) constructions. For instance, ***cut corners*** is a MWE but *trim corners* and *cut edges* are not, i.e., the semantic similarity (often modelled by word embeddings) between *cut* and *trim* or *corners* and *edges* provides few hints for correctly distinguishing a MWE from non-MWEs. This implies *strong lexicalization*, that is, it is the combination of precise lexemes (and not so much of their senses) which makes a MWE. Savary et al. (2019a) claim that, due to these properties, the generalization power of mainstream machine learning is relatively weak for MWE identification. This fact is confirmed by the results in the present paper, in which we outperform learning-based state-of-the-art systems using simple and interpretable rules and filters.

Savary et al. (2019a) show the critical difficulty of unseen data in MWE identification, whatever the system's architecture. They also argue that there is room for improvement on seen MWEs occurring with morphosyntactic variation. This fact, together with the dominance of seen data mentioned in Table 1, implies that much improvement can be achieved in MWE identification by focusing on seen MWEs. Another important finding for our proposal is that literal readings of MWEs, as in (3), occur surprisingly rarely in texts across languages from different genera. Savary et al. (2019b) show that, whenever an MWE's lexicalized components co-occur fulfilling the morphosyntactic conditions for an idiomatic reading, this reading almost always occurs. These findings inspired our method. We believe that competitive results should be achievable for seen MWE identification based on the following hypotheses:

- H1 We should search co-occurrences of precise lexemes (and POS) annotated as VMWEs in `train`.
- H2 We should allow only for those morphosyntactic variants which were previously seen in `train`.
- H3 We should not heavily rely on automatic POS tagging and syntactic parsing, which may be noisy.
- H4 We should consider syntactic coherence to eliminate coincidental occurrences as in (4).

[6] This definition was updated in edition 1.2, so that a VMWE from `test` is considered seen if it occurs in `train` *or* `dev` (vs. only in `train` for edition 1.1). Since we compare our results with those of edition 1.1, we use the `train`-only definition.

[7] The notions of MWE tokens and types are detailed in the framework proposed by Savary et al. (2019a), Sec. 2.

These hypotheses point towards simple *extraction* and *filtering* techniques. We put forward a method based on 8 fully interpretable filters directly inspired by the nature of MWEs discussed above. It only has 8 binary parameters, which indicate if, for a given language, a given filter should be activated or not. Its results are fully interpretable since it is straightforward to point at the filters which keep/eliminate a given true/false positive/negative candidate. Despite this simplicity and interpretability, our method outperforms the state-of-the-art systems based on complex architectures, including advanced machine/deep learning techniques, and requiring up to millions of parameters. We also expect the method to generalize over many languages since the properties it exploits proved generic by the state-of-the-art studies. Details of this method and an illustration of the whole set of filters are exposed in the next section.

Extraction and filtering techniques have been employed in the past (Constant et al., 2017, Sec. 3.2.1). Dictionary lookup was performed using lemmas, POS and distance filters prior to machine translation (Carpuat and Diab, 2010; Ramisch et al., 2013). In bilingual MWE lexicon discovery, filters can be applied before (Bouamor et al., 2012) or after extraction (Caseli et al., 2010; Semmar, 2018). Filters can be implemented using finite-state machines (Silberztein, 1997; Savary, 2009) and parameterized using classifiers (Pasquer et al., 2018b; Pasquer et al., 2020a). Our originality lies in (a) proposing a highly interpretable method based on on/off filters, (b) evaluating it on several categories of seen VMWEs in 11 languages, and (c) outperforming less interpretable state-of-the-art machine learning models.

4 Filter-Based *Seen2020* System

Our system, called $Seen2020$, operates in the closed track and focuses on the seen VMWE identification task with a two-step approach: candidate (i.e. potential VMWE) *extraction*, followed by candidate *filtering*. This method will be illustrated with examples in French. First, the *extraction* phase provides VMWE candidates in `dev` (while training) or `test` (while testing) based on the previously seen lemma multisets (so as to follow hypothesis H1 from Sec. 3). Multisets allow extracting candidates in any order (as opposed to tuples) while keeping track of repeated lemmas (as opposed to sets). Assume that a French corpus contains `train` sentences (5-9) and `test` sentences (10-19).[8] Focusing on (FR) ***faire***$_{\text{VERB}}$ ***la***$_{\text{DET}}$ ***lumière***$_{\text{NOUN}}$ '*to make the light*'⇒'to shed light' seen once in `train` (5), the (non-exhaustive) list of candidates in Tab. 2 would be extracted, as they contain the three lemmas *faire*, *la* and *lumière*.

The extraction guarantees high recall at the expense of precision, e.g. $R = 1$ but $P = 0.013$ in Italian. Thus, more than half candidates in Tab. 2 are no VMWEs but coincidental (11–12;15-a,b,c) or literal (19) occurrences. Many spurious candidates may be found in the same sentence (15) because of frequent determiners like *la* '*the*'.[9] The extraction recall is not perfect either: Ex. (10) is not extracted since the determiner *la* disappears in the negative form. Errors in automatic lemmatization also affect recall.

`train`

(5) ***La***$_{\text{DET}}$ ***lumière***$_{\text{NOUN.sing}}$ *sera*$_{\text{AUX}}$ ***faite***$_{\text{VERB}}$ *sur ce drame.* '*The light will be done on this drama*'⇒'Light will be shed on this drama.' (VID)
(6) ***La***$_{\text{DET}}$ ***porte***$_{\text{NOUN}}$ *a*$_{\text{AUX}}$ *résolument*$_{\text{ADV}}$ *été*$_{\text{AUX}}$ ***fermée***$_{\text{VERB}}$ *aux initiatives.* 'The door was firmly closed on initiatives.' (VID)
(7) *Il* ***ferme***$_{\text{VERB}}$ ***la***$_{\text{DET}}$ ***porte***$_{\text{NOUN.sing}}$ *à une loi.* 'He closes the door on a law.' (VID)
(8) *Dorothée [...]* ***prit***$_{\text{VERB}}$ ***la***$_{\text{DET}}$ ***fuite***$_{\text{NOUN}}$ '*Dorothée took escape*'⇒'Dorothée absconded.' (LVC.full)
(9) *Le* ***fossé*** *entre les riches et les pauvres* ***se creusait.*** 'The gap between the rich and the poor widened.' (VID)

`test`

(10) *L'enquête n'a pas* ***fait*** *de* ***lumière*** *sur les causes du sinistre.* 'The inquiry shed no light on the disaster's causes.'
(11) *Sa lumière la fait briller.* 'Its light makes it shine.'
(12) *Celui-ci met en lumière le constat fait plus haut.* 'This one puts into light the observation made above.'
(13) ***La lumière*** *a enfin été* ***faite*** *sur ce drame.* 'The light was finally shed on this drama.'
(14) ***Faire*** *une partie de* ***la lumière*** *sur le projet de réforme.* 'To shed part of the light on this reform project.'
(15) [...] *la lumière nocturne, qui n'a pas fait l'objet d'autant de recherches que la lumière diurne.* '[...] the nocturnal light, which has not been subject to as much research as the diurnal light.'
(16) *Une enquête a été ouverte pour* ***faire les lumières*** *sur ce drame.* 'An inquiry was launched to shed the lights on this drama.'
(17) ***La lumière*** *sur son rôle dans cette affaire doit être* ***faite.*** 'The light on its role in this case must be shed.'
(18) ***La lumière*** *est* ***faite*** *sur ce drame.* 'The light is shed on this drama.'
(19) *La lumière est faite de diode LED.* 'The light is made of LED diodes.'

[8] These examples are purely illustrative, we did not use the PARSEME shared task `test` corpus to design the filters: Ex. (6) and Ex. (9) come from `train` and the rest from the Web.

[9] The lemma of *la* is *le*. It has other surface forms, such as *l'* in Ex. (15-b) or *le* in Ex. (12).

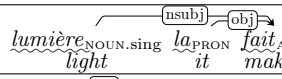
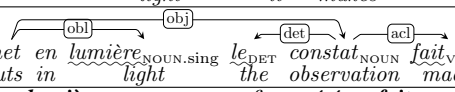
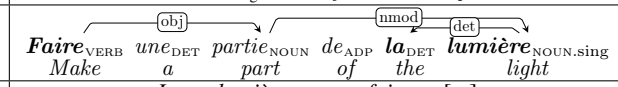
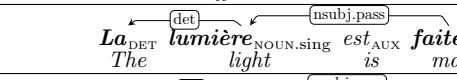
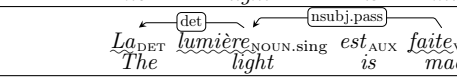
Example	Candidate	f1	f2	f3	f4	f5	f6	f7b	f8	$\sum_{i=1}^{8}$ fi
(11)			✓		✓	✓	✓	✓	✓	
(12)		✓			✓	✓		✓	✓	
(13)	***La***$_{\text{DET}}$ ***lumière***$_{\text{NOUN.sing}}$ a_{AUX} $enfin_{\text{ADV}}$ $été_{\text{AUX}}$ ***faite***$_{\text{VERB}}$ [...]	✓	✓	✓	✓	✓	✓	✓	✓	✓
(14)		✓	✓		✓	✓		✓	✓	
(15-a)	La_{DET} $lumière_{\text{NOUN.sing}}$ $fait_{\text{VERB}}$ [...]	✓	✓		✓	✓		✓	✓	
(15-b)	[...] $lumière_{\text{NOUN.sing}}$ $fait_{\text{VERB}}$ l'_{DET}[...]				✓	✓		✓	✓	
(15-c)	[...] $fait_{\text{VERB}}$ [...] la_{DET} $lumière_{\text{NOUN.sing}}$ [...]	✓	✓	✓				✓	✓	
(16)	[...] ***faire***$_{\text{VERB}}$ ***les***$_{\text{DET}}$ ***lumières***$_{\text{NOUN.plur}}$ [...]	✓	✓	✓	✓	✓	✓		✓	
(17)	***La***$_{\text{DET}}$ ***lumière***$_{\text{NOUN.sing}}$ sur_{ADP} son_{DET} $rôle_{\text{NOUN}}$ $dans_{\text{ADP}}$ $cette_{\text{DET}}$ $affaire_{\text{NOUN}}$ $doit_{\text{VERB}}$ $être_{\text{AUX}}$ ***faite***$_{\text{VERB}}$	✓	✓			✓	✓	✓	✓	
(18)		✓	✓	✓	✓	✓	✓	✓	✓	✓
(19)		✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 2: Sample candidates for the VMWE ***faire la lumière*** in Ex. (11-19). The ✓symbol means that the candidate is kept by a specific filter `f` (see Sec. 4).

Second, the *filtering* phase of *Seen*2020 aims at increasing precision. To this end, 8 filters `f1` to `f8` take morphosyntactic properties of VMWE components into account.

`[f1]` Components should be disambiguated: Lemmas may be shared by different words, as in $light_{\text{NOUN}}$ vs. $light_{\text{ADJ}}$ and should be disambiguated by their POS. Filter `f1`, inspired by hypothesis H1, only retains the candidates with the same POS multiset as the seen VMWE, e.g. {VERB,DET,NOUN} for ***faire la lumière***. However, as suggested by hypothesis H3, we handle some POS variations observed in the given VMWE in `train`, e.g. AUX vs. VERB.[10] In Tab. 2, `f1` selects (12-19) but excludes (11) whose POS multiset {AUX,PRON,NOUN} does not match the required multiset {VERB,DET,NOUN}. One drawback of `f1` is not to individually match each lemma and its POS. Thus, *it lights*$_{\text{VERB}}$ *the shed*$_{\text{NOUN}}$, would be extracted based on *it sheds*$_{\text{VERB}}$ *the light*$_{\text{NOUN}}$. Although such cases are relatively rare, we would like to replace POS multisets by ⟨lemma-POS⟩ multisets in the implementation of `f1` in the future.

`[f2-f3]` Components should appear in specific orders: With `f2` and `f3`, inspired by H2, we approximate the allowed syntactic transformations by the ordered sequence of POS. In French for instance, the passivization, illustrated by (5): (i) requires the noun to appear before the verb and (ii) often implies an inserted auxiliary. We split both aspects into specific filters: either we only look at the ordered POS sequence of the lexicalized components disregarding discontinuities (`f2`), which allows for more generalisable sequences,[11] or we also consider discontinuities (`f3`), which appears as more reliable,[12] but limited by the corpus' representativeness. In both cases, we check whether the candidate's ordered POS sequence has already been observed for any VMWE having the same POS multiset (for `f2`) and belonging to the same VMWE category (for `f3`), supposing that some allowed syntactic transformations may depend on the VMWE categories. We illustrate `f2` and `f3` with the examples in Tab. 2:

- The POS multiset in ***faire la lumière*** in (5) is {VERB,DET,NOUN}, as in ***fermer la porte*** '*to close the door*' (6-7) and ***prendre la fuite*** '*to take the escape*'⇒'to abscond' (8). For `f2`, we can thus compare (11-19) in `test` with (5-8) in `train`. The latter suggest that the POS multiset {VERB,DET,NOUN} only tolerates two POS sequences: **DET-NOUN-VERB**, as in (5-6) and **VERB-DET-NOUN**, as in (7-8), thus excluding, in Tab. 2, NOUN-DET-VERB in (12) or NOUN-VERB-DET in (15-b).

[10] For instance, (FR) ***se faire*** occurs 13 times with *faire* as a VERB (*les restaurants ouverts* ***se faisaient***$_{\text{VERB}}$ *rares* '*open restaurants were getting scarce*') and 5 times as an AUX (*Barbie* ***se fait***$_{\text{AUX}}$ *virer du tournage* '*Barbie is fired from the shooting*').

[11] This filter is indeed insensitive to the variety of discontinuities that could be associated with the sequence **VERB-DET-NOUN** in `train`, such as **VERB**-ADV-**DET**-ADJ-**NOUN**, **VERB-DET**-ADJ-**NOUN**, **VERB**-ADV-**DET-NOUN**, etc.

[12] It would for instance eliminate candidates like *Il fait éteindre*$_{verb}$ *la lumière* '*He makes turn off the light*' since the insertion of a verb between the lexicalized verb and determiner was never observed in `train`.

- The category of ***faire la lumière*** is VID. Other VIDs in `train` with the same POS multiset are (6-7) but not (8). For `f3` we thus compare (11-19) in `test` with (5-7) in `train`. There, the relevant POS sequences including discontinuities are **DET-NOUN**-AUX-**VERB** in (5), **DET-NOUN**-AUX-ADV-AUX-**VERB** in (6) and **VERB-DET-NOUN** in (7). Among the candidates kept by `f3`, we find true positives (18) but also false positives mainly due to literal readings (19). Ex. (17) is wrongly eliminated, since its sequence of discontinuities has never been observed in `train`. This highlights the sensitivity of `f3` to the training data, even though most VMWEs occur with few insertions (94% with a discontinuity length lower than 3 in the French corpus), which limits this coverage problem.

Like in `f1`, a drawback in `f2`–`f3` is to examine the POS sequences independently of the lemmas, which may (rarely) affect VMWEs with repeated POS, like (FR) ***donner***$_{\textbf{VERB}}$ ***sa***$_{\textbf{DET}}$ ***langue***$_{\textbf{NOUN}}$ ***au***$_{\textbf{ADP+DET}}$ ***chat***$_{\textbf{NOUN}}$ '*to give one's tongue to the cat*'⇒'to give up'.

[`f4`] Components should not be too far: In the French corpus, 86% of VMWE components are contiguous or only separated by one element. Therefore `f4` excludes candidates whose discontinuity length is higher than the highest length observed in `train` (excluding hapaxes) for the given category. Since long-distance dependencies are rare and their parsing is error-prone, `f4` approximates syntactic coherence of the extracted candidates (hypothesis H4), while at the same time overcoming probable parsing errors (H3) by simply ignoring the parsing trees. Here, the highest discontinuity length for VIDs in the French corpus is 6 in (9), which eliminates (15-c) and, wrongly, (17), that respectively have 7 and 8 inserted elements. The scarcity of the training data is a limit for this filter.

[`f5`] Closer components are preferred over distant ones: With `f4`, we limit discontinuity to the highest observable length, but most VMWEs are (quasi-)continuous. In the French corpus, 1,334 VIDs are continuous while only 2 VIDs have 6 insertions. Therefore, `f5` reinforces `f4` by favouring candidates with the lowest discontinuity length in each sentence, which further approximates syntactic coherence (H4) and insensitivity to parsing errors, as syntactic trees are ignored in `f5` (H3). For instance, (15) yields 3 candidates (15-a,b,c) due to the repetition of the noun and the determiner. The discontinuity length is 6 for (15-a,b) and 7 for (15-c), therefore only the former are kept.

[`f6`] Components should be syntactically connected: In accordance with H4, we expect the components of a VMWE to be most often syntactically connected. Thus, `f6` only retains those candidates which have: (i) more than 2 components and form a connected dependency subtree, (ii) 2 components connected by a syntactic chain with up to one insertion. This eliminates coincidental occurrences like (12) and (15-b,c) but keeps most VMWEs, including those with complex determiners, as in ***take*** $\overset{\text{obj}}{\frown}$ *a number* $\overset{\text{nmod}}{\frown}$ *of* ***decisions***. Candidate (14) also contains a complex determiner but is eliminated since it includes more than 2 components. Non-VMWEs wrongly kept by `f6` include literal readings like (19).

[`f7`] Nominal components should appear with a seen inflection: As suggested by H2, `f7` focuses on the nominal inflectional inflexibility, which is relevant for many VMWEs, especially idioms (Fazly et al., 2009). A candidate is kept by `f7` only if: (i) it contains a unique noun whose inflection is the same as in the VMWE seen in train, or (ii) it contains zero or more than one noun. This filter has 2 language-dependent versions. For languages whose nouns in VMWEs are marked for case (DE, EL, EU, HU, PL and RO),[13] filter `f7a` only checks the noun's case inflection. For other languages (BG, FR, HE, IT and PT), the number inflection is controlled by `f7b`. In Tab. 2, the singular inflection of *lumière* '*light*' observed in `train` is respected by all candidates, except (16), which is however a VMWE. This illustrates again the limits due to data sparcity.

[`f8`] Nested VMWEs should be annotated as in `train`: The extraction procedure may yield embedded candidates, some of which may be spurious. For instance, the VMWE ***il y a lieu*** '*it there has place*'⇒'one should' contains another VMWE ***il y a*** '*it there has*'⇒'there is', as well as a literal occurrence of ***a lieu*** '*has place*'⇒'takes place'. Arguably, all occurrences of the outermost VMWE should have the same nested annotation, whatever the context. Therefore, filter `f8` mimics the annotation done in `train` to decide whether to keep embedded candidates. The VMWE ***faire la lumière*** has never been seen as embedded in another VMWE in `train`, therefore all candidates in Tab. 2 are kept.

[13] We then require the proportion of VMWEs with noun case information to be higher than 50% of the total, since some VMWEs are likely to include foreign words with case.

As highlighted in the last column of Tab. 2, the activation of all 8 filters is not enough to distinguish VMWEs from non-VMWEs, since it results both in false positives (19) and in true negatives (14;16-17). The overall goal is, thus, to determine, for each language, the optimal set of filters to activate on `test` among the $2^8 = 256$ combinations. To achieve this, we consider as the best combination the one that maximizes the seen VMWE identification F-score obtained on `dev`, as described in Section 5.

5 Results

We perform experiments on 11 languages of the PARSEME shared task 1.1. The `dev` corpora are used for tuning the 8 parameters, i.e. choosing the optimal combination (among 2^8=256 combinations) of activated filters for each language. The best combination of filters is evaluated on the `test` corpora, and compared with the state-of-the-art systems trained and evaluated on the same data, as described is Sec. 2.

5.1 Parameter Tuning

Figure 1 shows, for FR, PL, PT and RO (see other languages in App. A), the seen VMWE identification F-score as defined in Ramisch et al. (2018), henceforth Seen-VMWEI F-score, on `dev` as a function of the 256 combinations. The focus on the left of the gap is on the 10 combinations with highest performances, while the curves to the right of the gap illustrate the decrease of performances with the 65 least relevant combinations. The gaps are placeholders for the 181 combinations with intermediate performances that were omitted for better readability. Scores decrease more in FR, PT and RO (by 58 to 75%) than in PL (15%).[14] The lowest scores are obtained in FR, PT and RO when few filters are activated (i.e. 1–5 black boxes occur in the rightmost column of the mosaic), whereas the best ones correspond to 3–5 activated filters. In PL, we note an opposite tendency: 3 activated filters lead to better results than 6. Some statistics and interpretation about the most frequently activated filters are proposed in Section 6. Note however that all languages reach a plateau for the best 10 combinations, which highlights the fact that the tuning is not that critical, and consequently allows some generalization. Once the best (i.e. the leftmost) combination has been found, it is applied on `test` for the final evaluation.

5.2 Comparison with the PARSEME Shared Task Systems

Fig. 2 compares the Seen-VMWEI F-score per language obtained on `test` by $Seen2020$ and 4 PARSEME shared task (edition 1.1) systems. $Seen2020$ (shown with a solid line) outperforms $VarIDE$ (which also specializes in seen VMWEs), gets better scores than $TRAPACC_S$ for 9/11 languages and ranks first for 7 languages.

Tab. 3 further compares macro-average scores on the 11 languages weighted according to the available data per language. *Seen2020* outperforms all 4 PARSEME shared task systems on the seen data (i.e. in Seen-VMWEI F-scores) and even on all (seen and unseen) data, although our system was not conceived to deal with unseen VMWEs at all and its F-score on unseen VMWEs is null. This shows that thorough account of seen VMWEs greatly boosts global MWE identification. In other words, with competing systems being extremely limited to deal with unseen VMWEs (Savary et al., 2019a), our method's superiority on (more frequent) seen VMWEs compensates the absence of treatment for unseen ones.

Comparison with $SHOMA$ 2019 is harder since it was evaluated outside the PARSEME shared task on 4 selected languages only.[15] On the 2 common languages, $SHOMA$ 2019 performs better than $Seen2020$ for German (F=0.80 vs. F=0.78) but worse for French (F=0.86 vs. F=0.88).

5.3 Error Analysis

Errors fall can be false negatives (affecting recall) and false positives (affecting precision). False negatives (additionally to unseen VMWEs, which, by the definition of the task, are never identified) are

[14] This is consistent with Savary et al. (2019b), who employ a procedure similar to ours for candidate extraction. Their number of pre-extracted candidates is 1.35 to 7.9 times lower in Polish than in Greek, German, Portuguese and Basque, despite similar amounts of annotated VMWEs. This shows that Polish has a lower potential for literal and coincidental occurrences, so our filters have a relatively small amount of false positives to filter out, which yields the relatively flat curve for PL in Fig. 1.

[15] Its code was openly published but we did not succeed in making it operational.

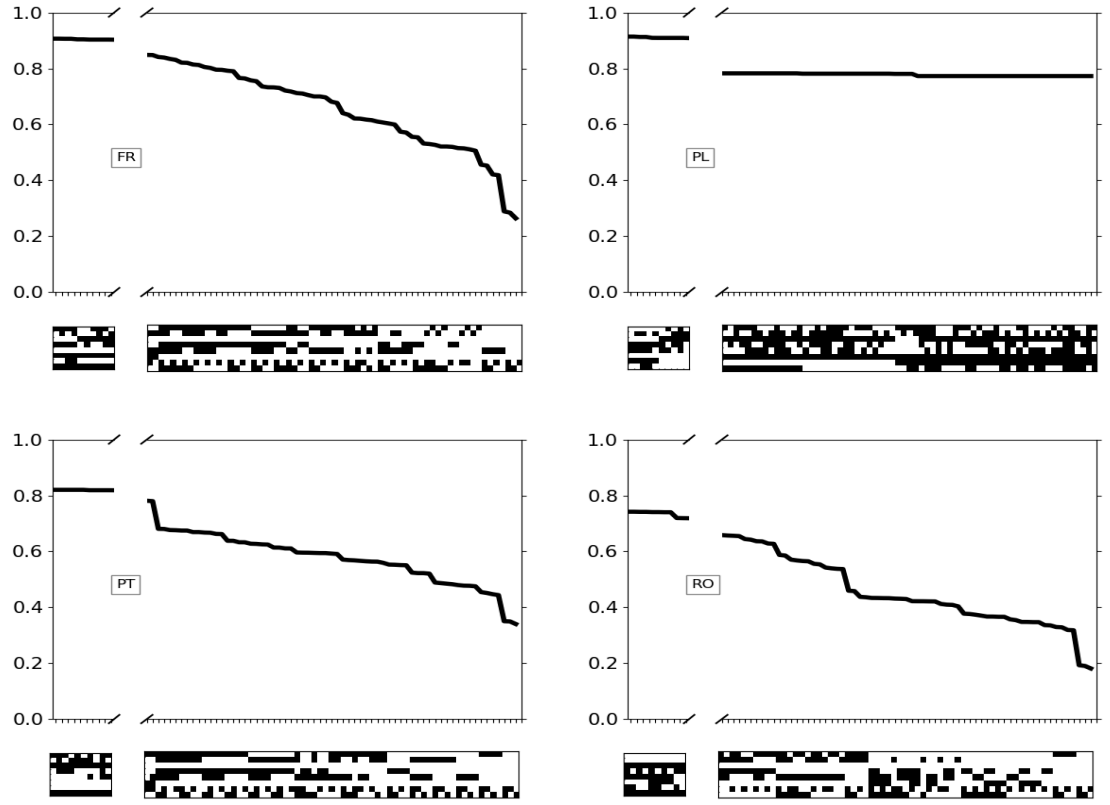


Figure 1: Seen-VMWEI F-score for FR, PL, PT and RO in `dev` as a function of the activated filters (in black) with `f1` (resp. `f8`) on the top (resp. at the bottom) of the mosaic. Only the 10 (resp. 65) configurations with highest (resp. lowest) F-score are represented to the left (resp. right) of the gap, e.g., the best FR score is obtained with `f1`, `f4`, `f6` and `f8` active.

notably due to: (i) inconsistent POS annotations, as in (FR) ***prendre au sérieux***$_{\text{ADJ/NOUN}}$ '*to take seriously*', and (ii) data scarcity, e.g. (PL) ***zrobienie zakupów***$_{Gen}$ 'doing shopping' was seen in `train` with the noun in genitive, while it occurs in `test` with accusative ***zrobić zakupy***$_{Acc}$ 'do shopping'.[16]

False positives have a variety of causes. First, the optimal selection of filters fails to eliminate some coincidental occurrences like (FR) *mener à bien sa politique* '*lead to good one's policy*'⇒'fulfil one's policy', where a variant of ***mener sa politique*** 'carry out one's policy' was wrongly identified. Second, when selecting the POS sequences allowed by `f2`, repeated POS are not distinguished although their order might be constrained: in (FR) ***s'***$_{\text{PRON}}$ ***y***$_{\text{PRON}}$ ***connaître***$_{\text{VERB}}$ '*self there know*'⇒'to be an expert', the pronoun *se* (resp. *y*) always comes first (resp. second), so candidates with the inverse order are necessarily spurious. Third, frequent pronouns and verbs, when inflected, can generate many spurious candidates be-

[16]Such case variations with gerunds are regular in Slavic languages and might be addressed by more fine-grained language-specific filters, at the expense of lesser genericity.

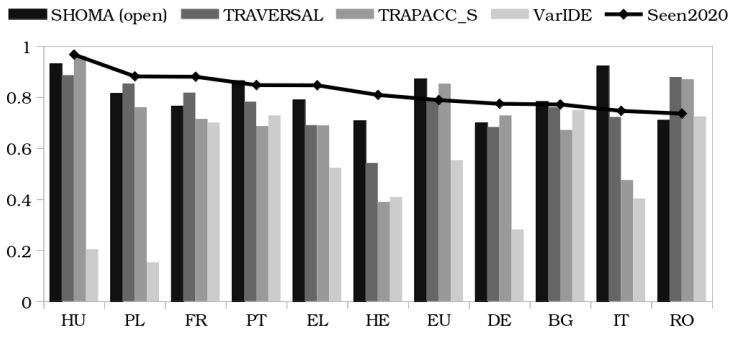


Figure 2: Seen-VMWEI F-score on `test` of *Seen*2020 vs. 4 PARSEME shared task systems as a function of the language

System	Macro-average F_1	
	seen	seen+unseen
SHOMA	0.81	0.64
TRAVERSAL	0.77	0.59
TRAPACC$_S$	0.73	0.57
VarIDE	0.61	0.49
Seen2020	**0.83**	**0.67**

Table 3: Macro-average F_1 scores for 4 PARSEME shared task systems and *Seen2020*.

cause their inflectional inflexibility is not covered by filters. For instance, (FR) *nous faisons* '*we make*' is wrongly marked as a variant of ***il fait*** '*it makes*'⇒'there is'. Third, some gold VMWE annotations are incomplete or missing. This explains the performance gap in RO between $Seen2020$ and $TRAVERSAL$ (0.74 vs. 0.88 for Seen-VMWEI F-scores): a standalone occurrence of the lemma *avea* '*to have*' wrongly annotated once as an LVC.full generates a drop of 26 points of F-score for the LVC.full category (0.32 vs. 0.58) Also, 22% of false positive IRVs are due to only one VMWE (***se poate*** '*oneself can*'⇒'it is likely'). Its verbal inflexion, here disregarded, could be used profitably, since this VMWE never occurs in plural while non-VMWE combinations of *sine* '*self*' with *poate* '*can*' in plural are frequent.

5.4 Last-Minute Results and Generalization

Shortly before submitting the final version of this paper, the results of edition 1.2 of the PARSEME shared task were announced.[17, 18] Seen2Seen (actually, $Seen2020$) scored best (out of 2) in the closed track and second (out of 9) across both tracks in terms of global MWE-based F-score. It outperformed 6 other open track systems, notably those using complex neural architectures and contextual word embeddings. It scored best (F=0.65), across both tracks in Italian, and second, with less than 0.01 point F-score difference behind the best (open track) system in Polish, Portuguese and Swedish (global F=0.82, F=0.73 and F=0.71). Also for phenomenon-specific measures Seen2Seen scored second across both tracks on both discontinuous and seen VMWEs. The only (open) system which outperformed Seen2Seen is a deep learning system using a generic multilingual BERT model (Devlin et al., 2019) tuned for joined parsing and VMWE identification. It scored a bit less than 0.04 F-measure point higher in the general ranking. Together with Seen2Seen, we submitted another system, Seen2Unseen, which relies on the former for seen VMWEs and adds discovery methods to cover unseen VMWE (Pasquer et al., 2020b).

6 Interpretability and Generalization

Our method proves encouraging. Not only does it outperform state-of-the-art systems, even those in the open track, but also it is interpretable. First, it is straightforward to identify the filters responsible for errors made by the system, enabling incremental development and customization to specific needs. Second, the filters are based on well known and pervasive linguistic properties of VMWEs. Third, these properties are generic enough to allow us to put forward cross-language interpretations and perspectives.

For instance, the differences observed between the corpora in FR, PT, RO on the one hand and PL on the other hand could be due to the fact that the filters were initially conceived for a Romance language (FR). Consequently, they may tend to perform better for languages from the same family than for Slavic ones, where fine case-number inter-dependencies occur in nouns (BG, even if Slavic, exhibits no case inflection). Also, our filters are less relevant to languages with a large number of single-token VMWEs, like HU (where our system's performances are still very high).

We may also observe interesting cross-language tendencies, such as a performance drop in BG, EL, EU, IT and PT at the beginning of the second part of Fig. 1 and 3. It occurs with the simultaneous activation of filters `f2`, `f4` and `f5` while `f3` and `f6` are inactive. This might be due to the fact that this configuration favors the nearest components without looking at their syntactic connection and uses `f2` which is less generic than `f3` to determine the allowed POS sequences.

Globally the most discriminative filters across all 11 corpora are: `f8` (embedded sequences) selected for most languages (8); `f4` (category-specific maximal linear distance between components) and `f6` (syntactic distance) for 7 languages; `f5` (minimal linear distance favored) for 6 languages; `f2` (sequence of POS of lexicalized components) for 4 languages; `f7` (noun inflection) for 3 languages and `f3` (sequence of POS including discontinuities) for 2 languages. `f1` (POS disambiguation) only appears in the best configuration for French. The single optimal cross-language configuration – based on the highest macro-averaged F-score on `dev` (F = 0.79) – is obtained when `f2`, `f4`, `f5` and `f6` are activated.[19] In other words, the distance between components, either linear or syntactic, appears as much more discrim-

[17] This subsection was not peer-reviewed.

[18] http://multiword.sourceforge.net/sharedtaskresults2020/

[19] The lowest score (F = 0.76), obtained when no filter is active, is 3 points lower than the best configuration per language.

inative than nominal fixity. This should however be tempered by the fact that not all VMWEs contain a unique noun, making `f7` much more specific than those filters based on distance.

The last-minute results mentioned in Sec. 5.4 show that our system generalizes well to new languages and remains competitive even when compared to contextual embedding models. It might be argued that its generalization power heavily depends on high-quality predicted POS tags and dependency trees, extensively used in filters, but potentially noisy (H3). However, the PARSEME corpora rely on Universal Dependencies (Nivre et al., 2020) and on the tools trained on them, whose quality is increasing. Recent results show POS tagging and parsing accuracies mostly exceed 90% and 80% for the languages of the PARSEME corpora.[20] Moreover, syntax is often approximated by POS sequences (in `f2`–`f5`) rather than used directly (`f6`). Finally, language-specific tuning of the filters mitigates low-quality parsing.

It could also be argued that filters such as `f7` could cover all types of inflection (e.g. verb tense, mood). We have extensively tested this hypothesis in previous work, automatically generating all possible feature-value combinations and selecting the most discriminant ones (Pasquer et al., 2020a). However, the results of this (computationally intensive) process were worse than those of $Seen2020$. More insight into generalization might stem from covering new languages and additional training data.

7 Conclusions and Future Work

We presented $Seen2020$, a system for the identification of seen VMWEs based on a combination of morphosyntactic filters. Based on state-of-the-art results, our method deliberately avoids complex machine learning and deep learning techniques for several reasons. Firstly, the type-level idiosyncrasy and strong lexicalization of MWEs (cf. Sec. 3), evidenced in many languages, suggest that distributional semantics might not be a strong ally in distinguishing MWEs from non-MWEs. This is confirmed by the recent results of deep-learning-based MWE identification systems, which – even if partly competitive with respect to the state of the art – still remain largely unsatisfactory. Secondly, the recent MWE identification results show: (i) the critical difficulty of generalizing over unseen data (ii) considerable room for improvement with respect to morphosyntactic variants of seen MWEs, (iii) the very low frequency of literal readings of MWEs. This means that focus on seen data and linguistic characterization of their variants should be enough to boost the global MWE identification results. Thirdly, we wish to avoid the complex architectures and non-interpretable results of neural methods.

Our system's main contributions are: (i) its simplicity since no complex tuning is necessary, (ii) higher performances (for 9/11 languages) than the best closed-track systems of a recent shared task and even higher than the best open-track system for 7 languages, (iii) ability to highlight linguistic properties of VMWEs, (iv) interpretability of the results and easy incremental development based on error analysis, (v) adaptability as new filters can be easily added. The system's main drawbacks are its high sensitivity to scarcity and the quality of the VMWE annotations (despite its ability to mitigate errors in other annotation layers, such as POS tagging or syntax).

As future work, we would like to further enhance our system's design, e.g. by representing VMWE as multisets of ⟨lemma-POS⟩ pairs rather than of lemma and POS multisets separately, which will help avoid some errors, as discussed above. Deeper per-language analyses of the results might also bring more insights on the generalization of our method and to the nature of VMWEs in general. For instance, we can study the generalization of our system on out-of-domain data, since we heavily rely on the lemmas of seen VMWEs, which may vary considerably across topics, registers and domains.

Acknowledgements

This work was funded by the French PARSEME-FR grant (ANR-14-CERA-0001). We are grateful to Guillaume Vidal for his prototype, and to the anonymous reviewers for their useful comments.

[20] http://ufal.mff.cuni.cz/udpipe/1/models

References

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing*, pages 267–292. CRC Press, Taylor and Francis Group, Boca Raton, FL, USA, 2 edition.

Dhouha Bouamor, Nasredine Semmar, and Pierre Zweigenbaum. 2012. Identifying bilingual multi-word expressions for statistical machine translation. In *Proc. of LREC 2012*, pages 674–679, Istanbul.

Marine Carpuat and Mona Diab. 2010. Task-based evaluation of multiword expressions: a pilot study in statistical machine translation. In *Proc. of NAACL/HLT 2010*, pages 242–245, Los Angeles, CA.

Helena de Medeiros Caseli, Carlos Ramisch, Maria das Gracas Volpe Nune, and Aline Villavicencio. 2010. Alignment-based extraction of multiword expressions. *Language Resources and Evaluation*, 44(1-2):59–77.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Afsaneh Fazly, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised Type and Token Identification of Idiomatic Expressions. *Computational Linguistics*, 35(1):61–103.

Ray Jackendoff. 1997. The architecture of the language faculty. *Linguistic Inquiry Monographs*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May. European Language Resources Association.

Caroline Pasquer, Carlos Ramisch, Agata Savary, and Jean-Yves Antoine. 2018a. VarIDE at PARSEME Shared Task 2018: Are Variants Really as Alike as Two Peas in a Pod? In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 283–289. Association for Computational Linguistics.

Caroline Pasquer, Agata Savary, Carlos Ramisch, and Jean-Yves Antoine. 2018b. If you've seen some, you've seen them all: Identifying variants of multiword expressions. In *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics*. The COLING 2018 Organizing Committee.

Caroline Pasquer, Agata Savary, Jean-Yves Antoine, Carlos Ramisch, Nicolas Labroche, and Arnaud Giacometti. 2020a. To be or not to be a verbal multiword expression: A quest for discriminating features. *arXiv preprint arXiv: 2007.11381*.

Caroline Pasquer, Agata Savary, Carlos Ramisch, and Jean-Yves Antoine. 2020b. Seen2Unseen at PARSEME Shared Task 2020: All Roads do not Lead to Unseen Verb-Noun VMWEs. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons (MWE-LEX 2020)*.

Carlos Ramisch, Laurent Besacier, and Alexander Kobzar. 2013. How hard is it to automatically translate phrasal verbs from English to French? In *MT Summit 2013 Workshop on Multi-word Units in Machine Translation and Translation Technology*, pages 53–61, Nice.

Carlos Ramisch, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, Tunga Güngör, Abdelati Hawwari, Uxoa Iñurrieta, Jolanta Kovalevskaitė, Simon Krek, Timm Lichte, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Behrang QasemiZadeh, Renata Ramisch, Nathan Schneider, Ivelina Stoyanova, Ashwini Vaidya, and Abigail Walsh. 2018. Edition 1.1 of the PARSEME Shared Task on Automatic Identification of Verbal Multiword Expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 222–240. ACL.

Omid Rohanian, Shiva Taslimipoor, Samaneh Kouchaki, Ruslan Mitkov, et al. 2019. Bridging the gap: Attending to discontinuity in identification of multiword expressions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2692–2698.

Agata Savary, Marie Candito, Verginica Barbu Mititelu, Eduard Bejček, Fabienne Cap, Slavomír Čéplö, Silvio Ricardo Cordeiro, Gülşen Eryiğit, Voula Giouli, Maarten van Gompel, Yaakov HaCohen-Kerner, Jolanta Kovalevskaitė, Simon Krek, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Lonneke van der Plas, Behrang QasemiZadeh, Carlos Ramisch, Federico Sangati, Ivelina Stoyanova, and Veronika Vincze. 2018. PARSEME multilingual corpus of verbal multiword expressions. In Stella Markantonatou, Carlos Ramisch, Agata Savary, and Veronika Vincze, editors, *Multiword expressions at length and in depth: Extended papers from the MWE 2017 workshop*, pages 87–147. Language Science Press., Berlin.

Agata Savary, Silvio Cordeiro, and Carlos Ramisch. 2019a. Without lexicons, multiword expression identification will never fly: A position statement. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 79–91, Florence, Italy, August. Association for Computational Linguistics.

Agata Savary, Silvio Ricardo Cordeiro, Timm Lichte, Carlos Ramisch, Uxoa I nurrieta, and Voula Giouli. 2019b. Literal Occurrences of Multiword Expressions: Rare Birds That Cause a Stir. *The Prague Bulletin of Mathematical Linguistics*, 112:5–54, April.

Agata Savary. 2009. Multiflex: A multilingual finite-state tool for multi-word units. In *Proc. of CIAA 2009*, pages 237–240, Sydney.

Nasredine Semmar. 2018. A hybrid approach for automatic extraction of bilingual multiword expressions from parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Max Silberztein. 1997. The lexical analysis of natural languages. In *Finite-State Language Processing*, pages 175–203. MIT Press.

Regina Stodden, Behrang Q. Zadeh, and Laura Kallmeyer. 2018. TRAPACC and TRAPACCS at PARSEME Shared Task 2018: Neural Transition Tagging of Verbal Multiword Expressions. In *LAW-MWE-CxG@COLING*.

Shiva Taslimipoor and Omid Rohanian. 2018. SHOMA at Parseme Shared Task on Automatic Identification of VMWEs: Neural Multiword Expression Tagging with High Generalisation. *CoRR*, abs/1809.03056.

Jakub Waszczuk. 2018. TRAVERSAL at PARSEME Shared Task 2018: Identification of Verbal Multiword Expressions Using a Discriminative Tree-Structured Model. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 275–282. Association for Computational Linguistics.

Appendix A. Performances for Other Languages

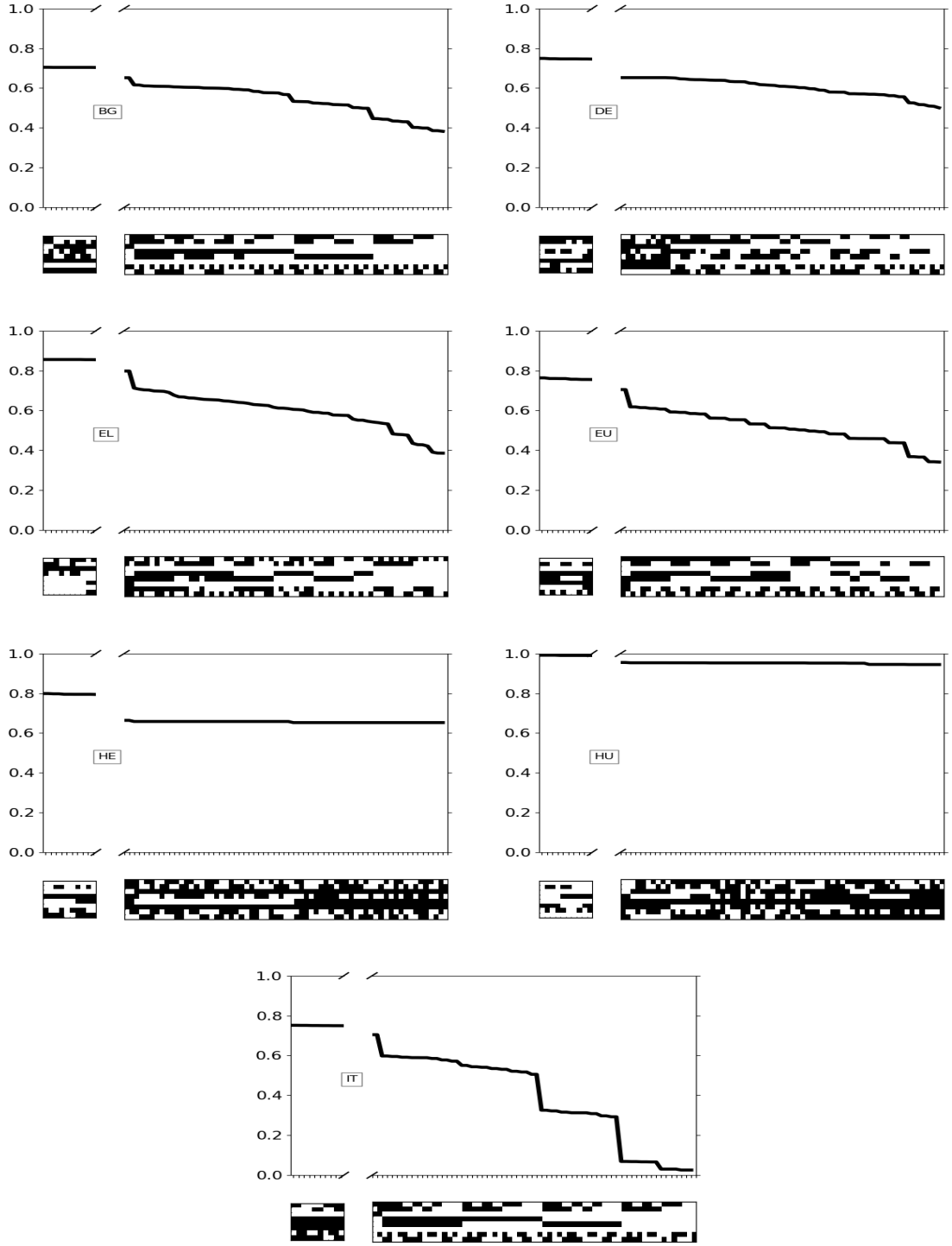


Figure 3: Task F-score in `dev` according to the activated `f1` to `f8` filters (in black) with `f1` (resp. `f8`) on the top (resp. at the bottom) of the mosaic. Only the 10 (resp. 65) configurations over 256 with higher (resp. lower) F-score are represented.