For a Fistful of Puns: Evaluating a Puns in Multiword Expressions Identification Algorithm Without Dedicated Dataset

Julien Bezançon and **Gaël Lejeune**
STIH/CERES, Sorbonne Université
28 rue Serpente, 75006 Paris
firstname.lastname@sorbonne-universite.fr

Abstract

Machine Translation systems have always struggled with challenges such as multiword expressions (MWEs) and wordplays. These phenomena are idiosyncratic, pervasive across languages, and significantly affect performance of MT systems (among others). In this context, we explore the nature of puns created from multiword expressions (PMWEs), characterized by the creation of a wordplay from a source MWE to recontextualize it or to give it a humorous touch. Little work has been done on PMWEs in NLP. To address this challenge, we introduce ASMR, an alignment-based PMWE identification and tagging algorithm. We offer an in-depth analysis of three different approaches to ASMR, each created to identify different types of PMWEs. In the absence of PMWE-related datasets and resources, we proceed to a snowclone detection task in English. We also perform a MWE identification task in 26 languages to evaluate ASMR performance across different languages. We show that ASMR exhibits state-of-the-art results for the snowclone detection task and produces interesting results with the MWE identification task. These results may indicate that ASMR is suitable for a PMWE identification task.

1 Introduction

A lot of work has been done on multiword expressions (MWEs) in NLP since their introduction to the field by Sag et al. (2002); Choueka (1988). They are generally described as combinations of words with a certain degree of idiomaticity at the lexical, syntactic, semantic, pragmatic and/or statistical levels (Baldwin and Kim, 2010). MWEs are usually non-compositional or semi-compositional (Gross, 1982), idiosyncratic, pervasive across different languages, and subject to varying degrees of variation (Ramisch, 2023). Other phenomena, such as ambiguity and discontiguity, may also be an issue (Constant et al., 2017). Because of these features, they represent a particular challenge in NLP, notably for Machine Translation systems, which need to take them into account (Zaninello and Birch, 2020).

Like any sequence of words, MWEs can serve as the basis for creating puns and other kinds of wordplays. Puns in multiword expressions (hereafter PMWEs) are characterized by the creation of a pun or a wordplay from a source multiword expression in order to recontextualize it or give it a humorous touch. By this process, MWEs such as (1) become (2) in the context of strikes in France in 2023.

1. "*l'heure est **grave***"
   (FR, it's a **serious** time)
2. "*l'heure est **grève***"
   (FR, it's a **strike** time)

Like MWEs, PMWEs can be translated from one language to another. For instance, (4) is a PMWE created from (3) working in both Italian and English. However, studies show that the translation of puns is not well handled by Machine Translation systems (Yu et al., 2018; Jiang et al., 2021).

3. "*l'alba dei morti **viventi***"
   (IT, **Dawn** of the dead, 1978)
4. "*l'alba dei morti **dementi***"
   (IT, **Shaun** of the dead, 2004)

To our knowledge, and unlike MWEs, PMWEs have not been extensively studied in NLP, with very few resources available and almost no dedicated work on them. We find that PMWEs can be interesting due to their dual nature as MWE and wordplay. Machine Translation tasks as well as Automatic Humor Analysis could benefit from their study. Moreover, PMWEs might be useful to study the morphosyntactic and semantic evolutions of MWEs, since they tend to accept new forms and/or meaning over time (Fiala and Habert, 1989). In some cases, they may even be completely replaced by one of their own PMWE (Cusimano, 2015).

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 8350–8370
November 4-9, 2025 ©2025 Association for Computational Linguistics

In addition to sharing the same difficulties as MWEs, PMWEs pose challenges of their own. Their identification in text can be even harder than that of MWEs, for several reasons: (i) they tend to be less frequent in texts than MWEs, (ii) although their source MWE generally remains recognizable, several letters or words may be modified when creating a PMWE and (iii) their meaning can be altered, making the use of semantic-based approaches more challenging. Finally, differentiating a PMWE from a MWE can be a complex task, even for an individual with a certain expertise in these entities, as shown in Bezançon et al. (2025b).

In this paper, we introduce ASMR (Align, Segment, Match, Rank), an alignment-based algorithm whose goal is to identify and tag PMWE candidates in texts. We first present the architecture of ASMR. We then proceed to various experiments in two different datasets in order to evaluate the performances of this algorithm.

Task 1 : Snowclone detection We evaluate how ASMR is able to detect snowclones (defined in Section 2) in a given set of sentences. To do so, we use the CATCHPHRASE dataset (Sweed and Shahaf, 2021).

Task 2 : MWE identification We aim to evaluate ASMR's ability to identify and tag MWEs in different languages by using the PARSEME 1.3 corpus (Savary et al., 2023), which consists of 26 languages. The PARSEME corpus, while lacking puns, is valuable as it includes MWE variants as well as discontinuous and unseen MWEs. Those subsets of MWEs share similarities with PMWEs (lexical variation, word order change, ...). In the absence of a dedicated dataset, we use PARSEME to evaluate ASMR's tagging capabilities and analyze case studies (Section 6).

With an older version of ASMR, we were able to identify PMWEs created from 216 MWEs in a corpus of French tweets (Bezançon and Lejeune, 2023). We were also able to identify a set of PMWEs created from formulas in Middle Arabic texts (Bezançon et al., 2025a). Both approaches rely on qualitative evaluation carried out by experts on a selection of N PMWE candidates. In the absence of a PMWE annotated dataset, we have not yet been able to evaluate the performances of ASMR from a quantitative perspective. By combining a snowclone detection task with a MWE identification and tagging task, we hope to gain a better insight into ASMR's functionalities. ASMR is available on GITHUB (AGPLv3 license), along with the scripts used for our experiments[1].

2 Related Work

MWE identification. The main focus of MWE processing in NLP is the identification task, whose goal is to tag MWEs from a lexicon or a list in a text. Direct string matching and rule-based methods such as the ones proposed by Stanković et al. (2016); Ramisch (2015) were the first approaches used to address this task and are still used to this day. More recent approaches use Large Language Models (LLMs) such as BERT (Devlin et al., 2019). In fact, LLMs-based methodologies tend to outperform other approaches for the task of MWE identification (Ramisch et al., 2020; Bui and Savary, 2024). For instance, Tanner and Hoffman (2023) use a rule-based pipeline along with a pretrained Bi-encoders for Word Sense Disambiguation (Blevins and Zettlemoyer, 2020). Taslimipoor et al. (2020) use a pretrained BERT model as well as a tree CRF architecture to tag verbal MWEs in the PARSEME 1.2 corpus. Swaminathan and Cook (2023) use multilingual LLMs to try to learn non-language-specific knowledge about MWEs and idiomaticity. Nevertheless, while pretrained LLMs seem to offer better results than more traditional approaches, they still have difficulties capturing their semantic aspect (Tayyar Madabushi et al., 2021; Zeng and Bhat, 2022). Wada et al. (2023) paraphrase MWEs to address this problem, demonstrating that taking into account relevant semantic information can help to identify MWEs. Since there are very few resources on PMWEs, approaches using language models seem all the more costly to implement. We therefore drew inspiration from rule-based approaches to design ASMR, using known properties of PMWEs to characterize and identify them. We also plan to implement some semantic information in our methodology.

Approximate String Matching. String matching consists in finding a sequence s within a set T. Approximate string matching, by contrast, aims to identify all sequences in T that are most similar to s (Hall and Dowling, 1980). Several approximate string matching algorithms exist, including the Boyer–Moore algorithm (Tarhio and Ukkonen, 1993), the Ukkonen algorithm (Ukkonen, 1993),

[1] https://github.com/JulienBez/ASMR

and Hamming distance (Hamming, 1950). Most of these algorithms rely on edit distance (Levenshtein, 1966), which measures the minimal number of operations (insertions, deletions, substitutions and permutations) required to transform one sequence into another. Given the nature of PMWEs, approximate string matching algorithms may provide valuable insights for their identification.

Wordplays Processing. Linguistic creativity, and therefore wordplays, are hard to deal with in NLP. As explained by Netzer et al. (2009); Saussure et al. (1949), humans tend to diversify their sets of relations between words, using cultural and emotional experiences for instance. As a result, the combinatorial possibilities for creating wordplays are almost infinite (Knospe et al., 2016). Few works report on wordplays detection. However, since 2022, the JOKER-CLEF participative task challenge teams of scientists on several wordplay detection tasks (Ermakova et al., 2022, 2023, 2024). Wordplay generation tasks, such as Valitutti et al. (2013), were also performed.

Snowclones Detection. A snowclone is generally illustrated by a prototypical form of a MWE with flexible positions ("X be the new Y", X and Y being the flexible positions). It is derived from a reference sentence ("**pink** is the new black", allegedly said by Gloria Vanderbilt in India, 1960) and used to create new forms ("**orange** is the new black", Netflix TV show, 2013). This notion was first coined by Geoffrey K. Pullum in a blog post from 2003[2]. Since then, snowclones have known a large set of definitions, often described as patterns that accept word substitutions (Liberman, 2006), taking up known and institutionalized MWEs that remain identifiable in all circumstances (Hill, 2018; Traugott et al., 2016). Hartmann and Ungerer (2023) propose a quantitative study of two snowclones, "X be the new Y" and "the mother of all X", by extracting new forms of these snowclones. While snowclones tend to be PMWEs, there is no saying that all PMWEs are snowclones. Snowclones correspond to patterns with predefined word substitution positions, but we argue that PMWEs do not necessarily comply with this rule (as for "may the force **bee** with you").

3 Introduction to ASMR

ASMR's main purpose is the identification and tagging of PMWEs. It can be described as an alignment-based, semi-supervised approach. ASMR takes a list of seeds, for instance prototypical forms of MWEs, as described in Pasquer (2019), and a list of sentences in which we want to identify PMWEs created from the seeds. As an output, ASMR creates a ranking of PMWE candidates for each seed. It consists of a succession of 4 processes, which we describe here. These processes are illustrated Table 3.

3.1 Alignment

First, ASMR creates alignments between each seed-sentence pairs. An alignment can be defined as the superposition of the elements of two sequences in order to highlight their similarities and differences. We give an example of alignment between two sequences in Table 1. We use the BIOPYTHON package[3] to create these alignments (see Appendix A). This package allows us to fetch multiple possible alignments for a given seed-sentence pair, as shown in Table 2.

May	the	-	**beer**	be	with	you
May	the	**force**	-	be	with	you

Table 1: Example of alignment at token level for the seed "May the force be with you" and the PMWE "May the beer be with you" (CATCHPHRASE dataset). In this example, the substitution of "force" by "beer" is highlighted by the misalignment between these tokens (in blue).

there s	no	place	like	long	island	no	place	like	home
there s	-	-	-	-	-	no	place	like	home
there s	no	place	like	-	-	-	-	-	home

Table 2: Two possible alignments between the seed "there's no place like home" and a sentence seen in the CATCHPHRASE dataset.

3.2 Segmentation

Once the alignments are made, we use them to find the longest common segment (LCS) between a seed and a sentence. This LCS will be our PMWE candidate. To find the LCS, we perform the following steps: (i) we retrieve each aligned token between a seed and a sentence and (ii) for each misalignment, we create a list containing all consecutive

[2] http://itre.cis.upenn.edu/myl/languagelog/archives/000061.html

[3] https://biopython.org/

Seed	some men just want to watch the world burn												
Sentence	some people really do just want to watch the world freeze												
Alignment	some	**men**	-	-	-	just	want	to	watch	the	world	**burn**	-
	some	-	**people**	**really**	**do**	just	want	to	watch	the	world	-	**freeze**
Segmentation	some	**[men]**				just	want	to	watch	the	world	**[burn]**	
	some	**[people,really,do]**				just	want	to	watch	the	world	**[freeze]**	
Match$_{exact}$	some					just	want	to	watch	the	world		
Match$_{fuzzy}$	some		**people**	**really**	**do**	just	want	to	watch	the	world		**freeze**
Match$_{combined}$	some		**people**			just	want	to	watch	the	world		**freeze**
Cand.$_{exact}$	some just want to watch the world												0.86
Cand.$_{fuzzy}$	some **people really do** just want to watch the world **freeze**												0.70
Cand.$_{combined}$	some **people** just want to watch the world **freeze**												0.80

Table 3: Alignment, segmentation, matching and resulting candidate (Cand.) for each approach for the seed "some men just want to watch the world burn" paired with the sentence "some people really just do want to watch the world freeze", found in the CATCHPHRASE dataset. For each seed-candidate pair, a cosine similarity is computed to rank candidates.

misaligned tokens, both for the seed and the sentence. We use these misalignment lists in the next step to match unseen tokens from the seed with substitute tokens from the corresponding sentence.

3.3 Matching

The matching process's goal is to isolate the LCS between a seed and a sentence. We provided 3 approaches to match tokens from the seed with tokens from the sentence, leading to the creation of 3 different approaches to ASMR: ASMR$_{exact}$, ASMR$_{fuzzy}$ and ASMR$_{combined}$.

ASMR$_{\mathbf{exact}}$ Only identical tokens between the seed and the sentence are matched. In other word, only the aligned tokens are matched, while the misaligned ones are ignored.

ASMR$_{\mathbf{fuzzy}}$ We match every single token between the first and the last common tokens between the aligned seed and sentence. If the X first tokens of the seed are unseen in the sentence, we match the first X tokens before the first common token in the sentence. We repeat this process with the Y last tokens of the seed: if they are unseen in the sentence, we match the Y first tokens after the last common token in the sentence.

ASMR$_{\mathbf{combined}}$ In addition to matching the aligned tokens between the seed and the sentence, we use misalignment lists to find the closest match for each unseen token from the seed. Let's take the following lists $list_{seed}$ and $list_{sent}$ from Table 3:

- $list_{seed}$ = [men]
- $list_{sent}$ = [people,really,do]

For each token tok_{seed} from $list_{seed}$, we compare its POS tag with the ones of each of the tokens in $list_{sent}$. The first token from $list_{sent}$ with the same POS tag is matched with tok_{seed}. If no token possesses the same POS tag as tok_{seed}, we compute a Levenshtein score between tok_{seed} and each token in $list_{sent}$ in order to find the best match. The only word from $list_{seed}$, "men", would therefore match with the first token of $list_{sent}$, "people", since they share the same POS tag.

Each approach was designed to provide a solution to a specific problem. ASMR$_{exact}$ can help us identify pun-free MWEs and provide a minimal tagging of MWEs. In contrast, it should not be able to find substitutes to unseen tokens in the seed, and therefore is most likely not suitable for PMWEs identification. ASMR$_{fuzzy}$, on the contrary, should be able to identify PMWEs, especially insertion and substitution based PMWEs, but will most probably produce a significant amount of noise, as it does not take discontinuity into consideration. Finally, ASMR$_{combined}$ will try to match the exact number of words seen in the seed by matching unseen tokens with substitutes. However, it should not be able to identify insertions.

3.4 Ranking

Prior to this step, we aligned, segmented and matched each seed with each sentence. As a result, we obtain a certain number of PMWE candidates for each seed. The final step of ASMR is to rank these candidates in order to sort them according to their probability of corresponding to a PMWE. We choose to use a cosine similarity score to rank the candidates for each seed. Other similarity measures could have been employed; however, comparing their performance lies beyond the scope of this paper. For experiments on this topic, see (Buscaldi et al., 2020; Koudoro-Parfait et al., 2021). We used

	Recall	Precision	F-score	Accuracy
ASMR$_{exact}$	**89±06**	73±14	79±09	89±10
ASMR$_{fuzzy}$	88±03	81±09	**84±05**	**93±03**
ASMR$_{combined}$	**89±02**	80±06	**84±03**	**93±02**
SVM (Sweed and Shahaf, 2021)	78±12	**84±13**	81±NA	85±08
RoBERTa (Sweed and Shahaf, 2021)	74±18	70±15	72±NA	81±94

Table 4: Results of ASMR for snowclone detection on the CATCHPHRASE test set. For the results of our approaches, the standard deviation is computed on 20 runs. Additionally, we manually computed F-scores for SVM and RoBERTa since (Sweed and Shahaf, 2021) did not report them.

the SCIKIT-LEARN implementation of the cosine similarity (formula 1 below).

$$s_c(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\| \vec{u} \| \| \vec{v} \|} \tag{1}$$

We compute a cosine similarity matrix between each seed u and all the PMWE candidates v extracted with this seed, as shown in 2.

$$M = \begin{bmatrix} s_c(\vec{u_1}, \vec{v_1}) & s_c(\vec{u_1}, \vec{v_2}) & \cdots & s_c(\vec{u_1}, \vec{v_n}) \\ s_c(\vec{u_2}, \vec{v_1}) & s_c(\vec{u_2}, \vec{v_2}) & \cdots & s_c(\vec{u_2}, \vec{v_n}) \\ \vdots & \vdots & \ddots & \vdots \\ s_c(\vec{u_m}, \vec{v_1}) & s_c(\vec{u_m}, \vec{v_2}) & \cdots & s_c(\vec{u_m}, \vec{v_n}) \end{bmatrix} \tag{2}$$

The ranking step can be repeated for numerous linguistic information layers. For instance, if our seeds and sentences are POS tagged, we can compute another similarity matrix between the POS tags of the seeds and the ones of the candidates. We argue that such process allows us to take into account various information in order to adjust our ranking of the candidates for each seed. In order to take all the available linguistic information layers into account at the same time, we calculate the mean similarity score of all layers for each candidate. Finally, we ponder our scores by taking into account the difference of length N between the seed and the candidate: if the candidate has X fewer tokens than the seed, we apply a rule of proportionality to its score S, as in 3.

$$S_{ponder} = \frac{S \cdot (N - X)}{N} \tag{3}$$

By applying this rule, we aim to discriminate candidates shorter than their seed, as a lot of them tend to be false positive. Additionally, shorter candidates that partially match the words of a seed tend to have better cosine similarity scores when compared with a seed, as seen for Cand.$_{exact}$ in Table 3.

4 Snowclone detection

We explained the features of ASMR. We now use the CATCHPHRASE dataset (Sweed and Shahaf, 2021) to evaluate ASMR capacity to detect if a sentence contains a snowclone.

Dataset. The CATCHPHRASE dataset consists of 3,855 snowclone-sentence pairs, of which 1,406 sentences allegedly contain the snowclone it was paired with. It proposes a binary classification task: for each snowclone-sentence pair, we must indicate whether the sentence contains the snowclone it was paired with. To achieve this classification task, Sweed and Shahaf (2021) used a Feature-based SVM model as well as a RoBERTa-based model. We report the recall, precision and accuracy they obtained with these models in Table 4. Surprisingly, their SVM model performed better than their RoBERTa model.

Parameters. As ASMR does not learn from input data, we use the train and dev partitions of CATCHPHRASE to determine the best parameters to run our experiments. We run ASMR with 240 distinct sets of parameters on the train partition (see Appendix B.2). These parameters include those of the vectorizer (number of ngrams and analyzer) and the threshold at which we consider a candidate to be a snowclone (according to its score). We only use token-level information during these runs. We select the 10 best sets of parameters for the train partition and the 10 best sets for the dev partition, for a total of 20 sets. We plan to run ASMR on the test partition with these 20 sets of parameters and to report standard deviation. We repeat this process for each approach, ASMR$_{exact}$, ASMR$_{fuzzy}$ and ASMR$_{combined}$ for a total of 720 runs.

Results. We report the results we obtained with ASMR with each approach in Table 4. ASMR$_{exact}$ obtained the best recall and ASMR$_{fuzzy}$ offered the best precision, as well as the best F-score and accuracy. ASMR$_{combined}$ achieves the best recall,

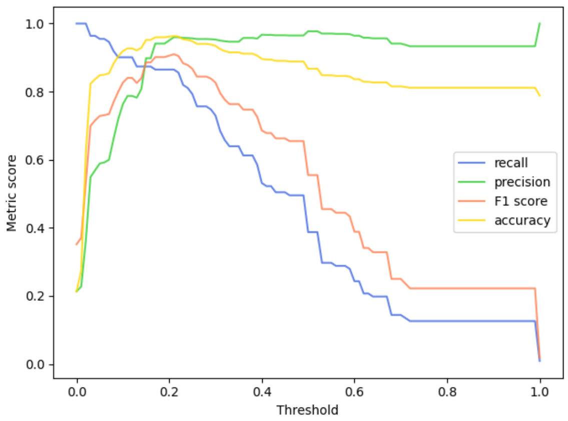

Figure 1: Impact of threshold on recall, precision, F-score and accuracy for the best run on the test partition of CATCHPHRASE with ASMR$_{combined}$.

F-score and accuracy. These last two matching algorithm might be slightly better than the first one due to the nature of snowclones, which are mainly created by substitution.

Overall, ASMR performs better than the models used by Sweed and Shahaf (2021) for the task of snowclone detection, although we note that our precision is slightly behind that obtained by their SVM model. Figure 1 shows the impact of threshold on the metrics we used for the best run (with ASMR$_{combined}$). As expected for a ranking system, the lower the threshold, the lower the precision and the higher the recall.

5 MWE identification

We measured the performance of ASMR for the task of snowclone detection in sentences with CATCHPHRASE. We now want to evaluate its ability to identify tokens belonging to MWEs in a given set of sentences.

Dataset. We use the version 1.3 of the PARSEME corpus (Savary et al., 2023), composed of 26 languages and mainly containing verbal MWEs. This corpus proposes a MWEs tagging task. So far, only 2 systems have been tested on the PARSEME 1.3 corpus: SEEN2SEEN (Pasquer et al., 2020) and MTLB-STRUCT (Taslimipoor et al., 2020). Savary et al. (2023) report the results for these 2 systems on PARSEME 1.3.

Parameters. The following steps are repeated for each language: (i) We retrieve a list of every MWEs seen in the train partition (lemmas and POS tags included). Since we collected lemmas for each word, we use them to align each MWE with each sentence. (ii) We run ASMR with 256 sets of parameters on the dev partition (see Appendix C.2). Those parameters consist of cosine similarity thresholds for the token layer, the morphosyntactic layer and the lemma layer. The possible thresholds were 0.1, 0.3, 0.7 and 1. We also compute a semantic score between each candidate and MWE using the SENTENCE-TRANSFORMERS package.This addition will enable us to assess the impact of semantic information on a MWE identification task using ASMR. Additionally, we remove candidates with discontinuities of more than 4 words. As shown in Pasquer (2019), the vast majority of discontinuous MWEs tend to have shorter discontinuities. (iii) We select the 10 best sets of parameters for the dev partition to run them on the test partition. We repeat this process for each approach with ASMR, totaling 768 runs per language. In the end, we performed 19,968 runs on PARSEME 1.3.

Results. Table 5 shows the global MWE-based results we obtained on the test set of PARSEME 1.3 for each language. Overall, ASMR$_{exact}$ obtained the best results among all the ASMR approaches, with a mean F-score of 52.6. Since ASMR$_{exact}$ only tag aligned words between a seed and a sentence, this result does not come as a surprise. ASMR$_{fuzzy}$ offers the best F-score for Hindi (HI), while ASMR$_{combined}$ obtained the best F-scores with Persian (FA) and Chinese (ZH). We report state-of-the-art results on the PARSEME 1.3 corpus with ASMR for Irish (GA), Croatian (HR) and Hindi (HI).

In order to analyze the impact of each feature used in ASMR, we generate boxplots for each feature and each threshold used with these features in Figure 2. These boxplots consist of F-scores obtained with every run made with ASMR$_{exact}$ on all languages on the dev partition. For instance, the first boxplot represents all the F-scores obtained with a threshold of 0.1 for the token feature. We observe that (i) regardless of the feature, a threshold of 1 seems to be too restrictive, as F-scores tend to be much lower, (ii) for the token and semantic features, we observe almost no variation with different thresholds, which can indicate that those features are not the most determinant for MWE identification with ASMR and (iii) the lemma and upos features show better F-scores with a threshold of 0.7, meaning that those features are probably the most helpful to identify MWEs with ASMR.

	ASMR$_{exact}$			ASMR$_{fuzzy}$			ASMR$_{combined}$			s2s
	R	P	F	R	P	F	R	P	F	F
AR	32.2±02	54.0±01	**40.3±01**	25.4±08	40.3±12	30.4±09	34.0±06	40.6±11	35.3±03	50.9
BG	72.1±01	55.3±00	**62.5±00**	61.9±11	56.8±04	58.4±04	63.3±10	57.4±03	59.5±03	65.7
CS	59.4±00	64.9±00	**62.0±00**	46.4±11	57.5±08	51.0±09	60.0±03	59.0±05	59.3±02	74.1
DE	20.7±00	67.3±03	**31.6±00**	16.4±03	38.8±19	22.2±06	18.6±01	43.3±15	25.5±03	71.4
EL	57.9±03	57.3±01	**57.5±01**	44.4±13	55.4±05	48.4±08	55.4±07	59.5±02	56.9±03	66.3
EN	44.4±01	78.0±00	**56.5±00**	32.4±08	66.7±15	42.6±08	42.8±03	72.1±08	53.6±03	59.9
ES	53.8±00	54.7±00	**54.2±00**	45.3±08	49.9±05	47.3±06	50.2±05	51.5±04	50.6±03	55.6
EU	72.7±01	76.4±03	**74.4±01**	62.3±10	74.5±07	67.2±07	71.3±05	69.1±08	69.8±04	82.1
FA	61.8±00	77.8±01	68.8±00	64.0±03	78.0±05	70.1±01	66.4±04	76.5±02	**71.0±02**	71.9
FR	66.2±04	73.6±01	**69.6±02**	50.2±13	57.5±13	53.5±13	65.9±04	65.1±07	65.2±03	78.7
GA	19.4±00	52.0±00	**28.2±00**	17.2±06	49.3±13	23.9±05	19.4±01	51.6±07	28.0±00	26.6
HE	35.8±01	64.1±01	**45.9±00**	33.9±02	53.6±10	41.3±04	36.3±01	57.4±06	44.4±02	46.9
HI	45.2±00	80.6±01	57.9±00	51.2±08	75.4±12	**59.6±02**	46.4±01	70.7±07	55.9±02	58.7
HR	64.1±01	91.9±00	**75.5±01**	49.7±13	77.9±14	60.1±13	61.9±03	79.8±09	69.5±04	75.3
HU	18.5±02	81.8±21	**29.4±01**	15.8±03	69.3±16	25.2±03	18.4±02	76.0±18	28.9±01	32.0
IT	59.0±01	64.0±01	**61.4±00**	50.0±07	55.2±09	52.2±07	58.6±02	61.1±03	59.8±01	65.0
LT	27.5±00	83.2±00	**41.3±00**	20.2±05	65.1±15	30.7±07	27.7±02	78.1±05	40.9±02	48.9
MT	14.2±02	19.2±01	**16.3±01**	16.0±04	16.4±03	15.7±02	10.4±04	15.2±04	12.1±04	16.5
PL	62.4±05	90.1±01	**73.6±03**	52.3±11	80.6±11	62.9±11	60.1±06	77.8±10	67.4±05	82.5
PT	51.4±07	70.0±07	**58.5±04**	34.6±05	47.3±07	39.3±03	53.4±10	59.0±07	54.8±05	74.0
RO	88.4±00	61.1±00	**72.3±00**	69.3±17	53.8±07	60.0±10	83.8±07	54.7±05	66.0±05	74.8
SL	51.2±04	33.2±01	**40.2±01**	33.7±16	29.2±04	30.0±09	49.7±04	30.2±03	37.4±02	41.8
SR	37.8±01	87.1±00	**52.7±01**	34.5±05	74.9±14	46.9±06	38.8±02	79.0±10	51.6±01	62.0
SV	29.2±01	80.8±03	**42.8±01**	25.1±04	70.1±13	36.7±05	28.4±02	74.2±03	41.0±02	82.2
TR	71.8±03	58.4±02	**64.4±01**	67.8±06	57.8±04	62.2±03	65.8±08	53.4±04	58.5±05	65.0
ZH	22.0±00	40.5±00	28.4±00	20.0±01	42.0±01	27.1±00	23.5±01	39.2±01	**29.2±01**	35.0
M	47.7	66	**52.6**	40	57.4	44.8	46.6	59.7	49.7	60.1

Table 5: Global MWE-based results on the test set of PARSEME 1.3 for 26 languages using ASMR. We report recall (R), precision (P), F-score (F) and mean (M) for all languages. Since we performed 10 runs for each language for our approaches, we also report the standard deviation. For the sake of comparison, we add SEEN2SEEN (s2s) system results. We highlight in bold the best F-score obtained with ASMR and underline state-of-the-art results.

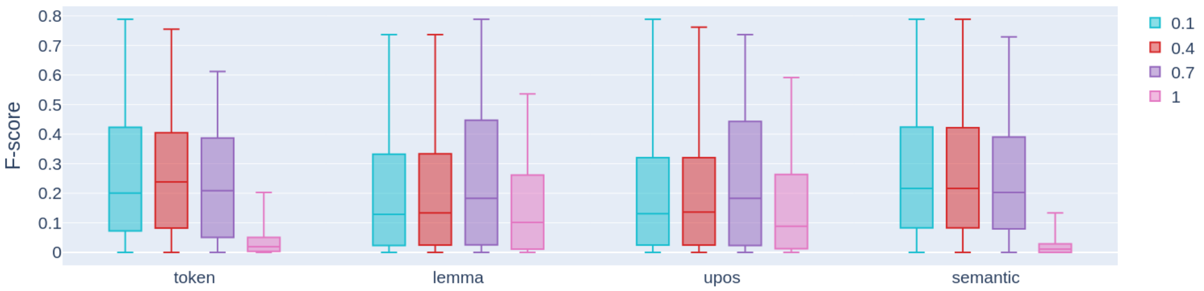

Figure 2: Boxplots of F-scores obtained on the dev partition of PARSEME 1.3 for each linguistic feature (token, lemma, upos and semantic similarity) for different thresholds (1, 0.7, 0.4 and 0.1) and each language. We used the F-scores obtained with ASMR$_{exact}$ since it has the best mean F-score among the 3 approaches we used.

6 Error analysis with PARSEME

Since PARSEME 1.3 offers several metrics on different subsets of MWEs, such as discontinuous and unseen ones, we can perform a more refined analysis of ASMR capabilities. Table 6 shows the mean F-scores across all languages on different subsets of MWEs. We observe that for two subsets (discontinuous and unseen-in-train) we achieve lower F-scores. Additionally, since ASMR was designed to identify PMWEs, we could argue that the Variant-of-train score is lower than expected.

	Exact	Fuzzy	Combined
Tok-based	55.0	48.1	52.6
Continuous	57.1	52.3	54.8
Discontinuous	**41.9**	**14.8**	**38.4**
Seen-in-train	68.0	61.0	66.9
Unseen-in-train	**00.9**	**06.8**	**05.2**
Variant-of-train	**60.1**	**50.3**	**59.4**
Identical-to-train	78.6	72.8	76.4

Table 6: mean F-scores across all languages obtained with each approach on different subsets of MWEs. We highlight the most interesting subsets (in **bold**).

Discontinuous. Discontinuous MWEs are a recurring challenge for MWE identification (Constant et al., 2017). As ASMR$_{fuzzy}$ and ASMR$_{combined}$ match misaligned words between a MWE and a candidate, low F-scores are expected. This is especially the case for ASMR$_{fuzzy}$, which match every word between the first and the last common words between a MWE and a candidate (as seen in Table 3). We observe that ASMR$_{exact}$, by tagging only aligned tokens, manage to obtain the highest mean F-score among the 3 approaches.

Unseen-in-train. One could argue that ASMR should be able to see a minimal number of unseen-in-train MWEs, especially with the ASMR$_{fuzzy}$ and ASMR$_{combined}$ approaches. We argue that this can be the case, notably with shorter, more generic MWEs, such as "break up". Table 7 shows 10 candidates found with ASMR$_{combined}$ for the MWE "break up". We observe the presence of other seen-in-train MWEs as well as 2 unseen-in-train MWEs. We also report erroneous candidates, which does not correspond to a MWE. While ASMR is capable of capturing both closely related MWEs and unseen MWEs, it might be difficult for it to distinguish good candidates from bad ones. This is highlighted by the ranking in Table 7, where seen, unseen and erroneous MWEs tend to blend together in the ranking.

Candidate	Cat	Tok	Upo	Lem	Sem	M
broke up	see	0.10	1.00	1.00	0.84	0.73
speak up	**uns**	0.55	1.00	0.48	0.45	0.62
fuck up	**uns**	0.20	1.00	0.21	0.44	0.46
look up	see	0.22	1.00	0.18	0.29	0.42
make up	see	0.12	1.00	0.11	0.41	0.41
ensure up	err	0.07	1.00	0.07	0.49	0.41
end up	see	0.03	1.00	0.03	0.54	0.40
jangle up	err	0.01	1.00	0.02	0.51	0.38
grow up	see	0.02	1.00	0.02	0.49	0.38
have up	err	0.02	1.00	0.03	0.46	0.38

Table 7: 10 ranked candidates for the MWE "break up". For each candidate, we report its score for each feature as well as its mean score (M, used for the ranking) and its subset (Cat). Possible categories are seen (see), unseen (uns) and erroneous (err).

Candidate	EN	Cat	M
получат помощ	get help	var	0.90
получиха помощ	get help	var	0.84
получават помощ	get help	var	0.73
каза помощ	say help	err	0.65
каза помощта	say help	err	0.63
поеха помощ	ask help	err	0.63
взе помощ	take help	var	0.61
стана помощ	become help	err	0.61
получи подкрепа	receive support	var	0.54
получи подкрепата	receive support	var	0.54

Table 8: 10 ranked candidates for the MWE "получа помощ" (BG, get help). For each candidate, we propose a minimal translation in english (EN) as well as its mean score (M) and its category (Cat). Possible categories are identical (idt), variant (var) and erroneous (err).

Variant-of-train. Variants of MWEs can correspond to several instances in the PARSEME 1.3 (see guidelines[4]). Among these instances, we find (i) syntactic variants, such as conjugated verb, change of tense or number and (ii) MWEs with some open slots (to take a decision). The former should be handled by morphosyntactic and lemmas analysis in most case, but the latter may have a direct impact on MWE identification, especially with ASMR. Table 8 shows 10 candidates found with ASMR$_{fuzzy}$ for the MWE "получа помощ" (BG, get help). We observe possible variations for both words of this MWE. "получа" can be conjugated and/or replaced by "взе" and "помощ" can be replaced by "подкрепата". Once again, the possible variations of this MWE blend with erroneous candidates in the ranking, making it hard to distinguish them. However, we observe that simple syntactic variants appear in the top candidates and therefore are easier to identify.

[4] https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.3/?page=010_Definitions_and_scope/030_Syntactic_variants_of_VMWEs

7 Discussion

In this work, we introduced ASMR, a Puns in MWE (PMWE) identification and tagging algorithm relying on sentence-level alignments and similarity scores to rank PMWE candidates. While earlier studies show that ASMR can be used to extract good PMWE candidates in both French and Arabic, no quantitative evaluation was yet performed, due to a lack of a PMWE annotated dataset.

To get around this issue, we proceeded to 2 experiments in order to evaluate ASMR functionalities. We first used a snowclone detection task on the CATCHPHRASE dataset in order to evaluate ASMR's capacity to assert the presence of a PMWE candidate in a sentence. We then used the PARSEME 1.3 corpus to evaluate ASMR identification and tagging performance on MWEs for 26 languages. We show that ASMR obtains state-of-the-art results on the snowclone detection task and for three languages with the MWE identification task (Irish, Croatian and Hindi).

We performed an in-depth analysis of the limitations we encountered with some subsets of MWEs within PARSEME, which allowed us to get a better understanding of ASMR performance. It showed that, while true positive and false positive candidates tend to blend together in the ranking, the top N candidates seem to be pertinent in most cases. This observation is highlighted by both the MWE identification task and the snowclone detection task, where higher thresholds lead to higher precision and lower recall. We also note that, while we performed multiple runs for each task, our standard deviations are low, which can account for the robustness of ASMR (see Appendix B.4).

We plan to create a PMWE dataset through participative sciences to further evaluate the performance of ASMR. Such dataset would also be useful to test the performance of other systems, either created for MWE or PMWE identification. We also plan to use ASMR to observe possible changes affecting MWEs over time. For example, the French expression "*être comme un poisson* ***hors de*** *l'eau*" ("to be like a fish **out of** the water") led to the creation of the PMWE "*être comme un poisson* ***dans*** *l'eau*" ("to be like a fish **in** water") in the 19th century (Fiala and Habert, 1989). Over time, such PMWEs can become conventionalized (as defined in Nunberg et al. (1994)) themselves, ultimately supplanting the original MWE (Cusimano, 2015).

ASMR offers a valuable tool for studying these diachronic processes. By tracking frequency patterns of canonical and variant forms, we can observe the emergence, transformation, and possible lexicalization of new MWEs in real time. Indeed, we have already observed such dynamics in the FRUIT corpus (Bezançon et al., 2025b): the original expression "*travailler plus pour gagner* ***plus***" (Nicolas Sarkozy, 2007, "work more to earn **more**") appears less frequently than the variant "*travailler plus pour gagner* ***moins***" ("work more to earn **less**"), suggesting an ongoing shift in usage and meaning.

Limitations

CATCHPHRASE experiment. We take into account several limitations, due to either the CATCHPHRASE dataset or the methodology we used: (i) the dataset itself is imbalanced. As stated by its authors, 64 % of the sentences do not contain the snowclone they were paired with (Sweed and Shahaf, 2021). (ii) the task doesn't evaluate the capacity of a system to tag tokens belonging to a snowclone. (iii) since CATCHPHRASE does not come with POS tag nor lemmas, we only tokenized both the snowclones and the sentences. (iv) the threshold itself can be seen as a limitation: the ideal threshold found for the train and dev partitions of the dataset might not always be the same for the test partition. Nevertheless, we find that for CATCHPHRASE, the ideal threshold seems to be roughly the same for all partitions (between 0.1 and 0.3).

PARSEME 1.3 experiment. To avoid overloading our calculation server, we had to limit the number of runs we made on the PARSEME 1.3 corpus. To limit this number, we did not manipulate the features of the vectorizer used to compute cosine similarity scores, which remained the same among all languages. We also limited to 4 the number of thresholds we used for each feature (using only thresholds of 0.1, 0.4, 0.7 and 1). Moreover, since ASMR was not initially designed to strictly identify MWEs, we added a rule to limit the size of possible discontinuities to 4. While this rule is also found in other systems, such as the one of Pasquer et al. (2020), we did not evaluate its impact on the MWE identification task with ASMR. Finally, ASMR does not account for phenomena such as permutation yet, which might have an impact on the results we obtained, since some MWEs allow word permutations.

Ethical considerations

We ran ASMR on an AMD EPYC MODEL 7543P MILAN 32 CORE CPU with 32GB of memory. We ran it on every language in parallel threads, for a cumulated time of 58 hours and a maximum time of 13 hours. We use this information along with the carbon intensity in France in 2024[5] to estimate our carbon footprint, which amounts to 120.45g estimated CO2 emission (or 0.12 kg). This estimation remains approximate, as we couldn't take every parameter into account. In comparison, Large Language Models such as BERT usually have a much higher carbon footprint (Wang et al., 2023).

Acknowledgments

We thank the SACADO unit of Sorbonne Université for allowing us to use their computing server to run our experiments on the PARSEME 1.3 corpus.

References

Timothy Baldwin and Su Nam Kim. 2010. *Multiword Expressions*, 2 edition. Chapman and Hall/CRC.

Julien Bezançon, Rimane Karam, and Gaël Lejeune. 2025a. Lost in variation: An unsupervised methodology for mining lexico-syntactic patterns in middle Arabic texts. In *Proceedings of the 4th Workshop on Arabic Corpus Linguistics (WACL-4)*, pages 25–37, Abu Dhabi, UAE. Association for Computational Linguistics.

Julien Bezançon and Gaël Lejeune. 2023. Reconnaissance de défigements dans des tweets en français par des mesures de similarité sur des alignements textuels. In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs*, pages 56–67, Paris, France. ATALA.

Julien Bezançon, Gaël Lejeune, Antoine Gautier, Marceau Hernandez, and Félix Alié. 2025b. Forbidden FRUIT is the sweetest: An annotated tweets corpus for French unfrozen idioms identification. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 70–86, Vienna, Austria. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

Van-Tuan Bui and Agata Savary. 2024. Cross-type French multiword expression identification with pretrained masked language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4198–4204, Torino, Italia. ELRA and ICCL.

Davide Buscaldi, Ghazi Felhi, Dhaou Ghoul, Joseph Le Roux, Gaël Lejeune, and Xudong Zhang. 2020. Calcul de similarité entre phrases: quelles mesures et quels descripteurs? In *Traitement Automatique des Langues Naturelles (TALN, 27e edition). Atelier DEfi Fouille de Textes*, pages 14–25. ATALA; AFCP.

Yaacov Choueka. 1988. Looking for needles in a haystack or locating interesting collocational expressions in large textual databases. In *User-Oriented Content-Based Text and Image Handling*, RIAO '88, pages 609–623, Paris, FRA.

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

Christophe Cusimano. 2015. Figement de séquences défigées. *Pratiques*, (159-160):69 78.

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. Association for Computational Linguistics.

Liana Ermakova, Anne-Gwenn Bosser, Tristan Miller, Victor Preciado, Grigori Sidorov, and Adam Jatowt. 2024. *Overview of the CLEF 2024 JOKER Track: Automatic Humour Analysis*, pages 165–182.

Liana Ermakova, Tristan Miller, Julien Boccou, Albin Digue, Aurianne Damoy, and Paul Campen. 2022. Overview of the clef 2022 joker task 2: translate wordplay in named entities. *Proceedings of the Working Notes of CLEF*, pages 1666–1680.

Liana Ermakova, Tristan Miller, Anne-Gwenn Bosser, Victor Manuel Palma Preciado, Grigori Sidorov, and Adam Jatowt. 2023. Overview of joker–clef-2023 track on automatic wordplay analysis. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 397–415. Springer.

Pierre Fiala and Benoît Habert. 1989. La langue de bois en éclat : les défigements dans les titres de presse quotidienne française. *Mots. Les langages du politique*, 21(1):83–99.

Maurice Gross. 1982. Une classification des phrases « figées » du français. *Revue québécoise de linguistique*, 11(2):151.

[5] https://www.sfen.org/rgn/2024-record-production-electricite/

Patrick A. V. Hall and Geoff R. Dowling. 1980. Approximate String Matching. *ACM Comput. Surv.*, 12(4):381–402.

R. W. Hamming. 1950. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160.

Stefan Hartmann and Tobias Ungerer. 2023. Attack of the snowclones: A corpus-based analysis of extravagant formulaic patterns. *Journal of Linguistics*, pages 1–36.

Ian E. J. Hill. 2018. Memes, munitions, and collective copia: The durability of the perpetual peace weapons snowclone. *Quarterly Journal of Speech*, 104(4):422–443.

Siyu Jiang, Zhiheng Zhang, Qiong Zhong, Jin Xie, and Weilin Wu. 2021. The system analysis and research based on pun recognition. *Journal of Physics: Conference Series*, 2044(1):012190.

Sebastian Knospe, Alexander Onysko, and Maik Goth. 2016. *Crossing Languages to Play with Words: Multidisciplinary Perspectives*. Walter de Gruyter GmbH & Co KG.

Caroline Koudoro-Parfait, Gaël Lejeune, and Glenn Roe. 2021. Spatial named entity recognition in literary texts: What is the influence of ocr noise? In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Geospatial Humanities*, pages 13–21.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707.

Mark Liberman. 2006. The proper treatment of snowclones in ordinary english.

S. B. Needleman and C. D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Yael Netzer, David Gabay, Yoav Goldberg, and Michael Elhadad. 2009. Gaiku : Generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 32–39, Boulder, Colorado. Association for Computational Linguistics.

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538. Publisher: Linguistic Society of America.

Caroline Pasquer. 2019. *Garder la trace, mettre de l'ordre et relier les points : modéliser la variation et l'ambiguïté des expressions polylexicales*. Phd thesis, Tours, France.

Caroline Pasquer, Agata Savary, Carlos Ramisch, and Jean-Yves Antoine. 2020. 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*, pages 124–129, online. Association for Computational Linguistics.

Carlos Ramisch. 2015. *Multiword Expressions Acquisition: A Generic and Open Framework*. Theory and Applications of Natural Language Processing. Springer International Publishing, Cham.

Carlos Ramisch. 2023. *Multiword expressions in computational linguistics*. thesis, Aix Marseille Université (AMU).

Carlos Ramisch, Agata Savary, Bruno Guillaume, Jakub Waszczuk, Marie Candito, Ashwini Vaidya, Verginica Barbu Mititelu, Archna Bhatia, Uxoa Iñurrieta, Voula Giouli, Tunga Güngör, Menghan Jiang, Timm Lichte, Chaya Liebeskind, Johanna Monti, Renata Ramisch, Sara Stymne, Abigail Walsh, and Hongzhi Xu. 2020. Edition 1.2 of the PARSEME shared task on semi-supervised identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 107–118, online. Association for Computational Linguistics.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. In *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, pages 1–15, Berlin, Heidelberg. Springer.

F. de Saussure, C. Bally, A. Riedlinger, and A. Sechehaye. 1949. *Cours de linguistique générale*. Payot, Paris.

Agata Savary, Cherifa Ben Khelil, Carlos Ramisch, Voula Giouli, Verginica Barbu Mititelu, Najet Hadj Mohamed, Cvetana Krstev, Chaya Liebeskind, Hongzhi Xu, Sara Stymne, Tunga Güngör, Thomas Pickard, Bruno Guillaume, Eduard Bejček, Archna Bhatia, Marie Candito, Polona Gantar, Uxoa Iñurrieta, Albert Gatt, Jolanta Kovalevskaite, Timm Lichte, Nikola Ljubešić, Johanna Monti, Carla Parra Escartín, Mehrnoush Shamsfard, Ivelina Stoyanova, Veronika Vincze, and Abigail Walsh. 2023. PARSEME corpus release 1.3. In *Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)*, pages 24–35, Dubrovnik, Croatia. Association for Computational Linguistics.

T. F. Smith and M. S. Waterman. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197.

Ranka Stanković, Cvetana Krstev, Ivan Obradović, Biljana Lazić, and Aleksandra Trtovac. 2016. Rule-based Automatic Multi-Word Term Extraction and Lemmatization. In *LREC*, pages 507–514, Portorož, Slovenia.

Raghuraman Swaminathan and Paul Cook. 2023. Token-level identification of multiword expressions using pre-trained multilingual language models. In

Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023), pages 1–6, Dubrovnik, Croatia. Association for Computational Linguistics.

Nir Sweed and Dafna Shahaf. 2021. Catchphrase: Automatic Detection of Cultural References. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1–7, Online. Association for Computational Linguistics.

Joshua Tanner and Jacob Hoffman. 2023. MWE as WSD: Solving Multiword Expression Identification with Word Sense Disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 181–193, Singapore. Association for Computational Linguistics.

Jorma Tarhio and Esko Ukkonen. 1993. Approximate Boyer–Moore String Matching. *SIAM Journal on Computing*, 22(2):243–260.

Shiva Taslimipoor, Sara Bahaadini, and Ekaterina Kochmar. 2020. MTLB-STRUCT @parseme 2020: Capturing unseen multiword expressions using multi-task learning and pre-trained masked language models. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 142–148, online. Association for Computational Linguistics.

Harish Tayyar Madabushi, Edward Gow-Smith, Carolina Scarton, and Aline Villavicencio. 2021. AStitchInLanguageModels: Dataset and methods for the exploration of idiomaticity in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3464–3477, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Elizabeth Closs Traugott, Graeme Trousdale, Elizabeth Closs Traugott, and Graeme Trousdale. 2016. *Constructionalization and Constructional Changes*. Oxford Studies in Diachronic and Historical Linguistics. Oxford University Press, Oxford, New York.

Esko Ukkonen. 1993. Approximate String-Matching over Suffix Trees. In *Proceedings of the 4th Annual Symposium on Combinatorial Pattern Matching*, CPM '93, pages 228–242, Berlin, Heidelberg. Springer-Verlag.

Alessandro Valitutti, Hannu Toivonen, Antoine Doucet, and Jukka M. Toivanen. 2013. "Let Everything Turn Well in Your Wife": Generation of Adult Humor Using Lexical Constraints. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 243–248, Sofia, Bulgaria. Association for Computational Linguistics.

Takashi Wada, Yuji Matsumoto, Timothy Baldwin, and Jey Han Lau. 2023. Unsupervised paraphrasing of multiword expressions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4732–4746, Toronto, Canada. Association for Computational Linguistics.

Xiaorong Wang, Clara Na, Emma Strubell, Sorelle Friedler, and Sasha Luccioni. 2023. Energy and carbon considerations of fine-tuning BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9058–9069, Singapore. Association for Computational Linguistics.

Zhiwei Yu, Jiwei Tan, and Xiaojun Wan. 2018. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1660, Melbourne, Australia. Association for Computational Linguistics.

Andrea Zaninello and Alexandra Birch. 2020. Multiword expression aware neural machine translation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3816–3825, Marseille, France. European Language Resources Association.

Ziheng Zeng and Suma Bhat. 2022. Getting bart to ride the idiomatic train: Learning to represent idiomatic expressions. *Transactions of the Association for Computational Linguistics*, 10:1120–1137.

A Alignments with Biopython

Biopython provides pairwise alignment methods to compare two sequences (originally DNA, RNA, or proteins sequences). These methods use dynamic programming algorithms such as Needleman-Wunsch (Needleman and Wunsch, 1970) for global alignment (which we used here) and Smith-Waterman (Smith and Waterman, 1981) for local alignment. We navigate through the alignments using list comprehension in Python, as described in Section 3.

B Snowclone detection task details

B.1 Catchphrase metadata

Table 9 shows some statistics on the Catchphrase dataset. Table 10 shows a sample of the Catchphrase dataset for two snowclones. For each snowclone, we report an identical match, a partial match and a mismatch.

	#Token	#Sentence	#Snowclone
train	50,292	2,974	1,235
dev	11,068	682	60
test	10,389	520	111
total	58,785	3,855	1,406

Table 9: Number of tokens, sentences and sentences containing a snowclone in Catchphrase.

Snowclone	Sentence	Label
may the force be with you	thank you and **may the force be with you**	1
may the force be with you	**may the gods be with you**	1
may the force be with you	the ache in my chest from not being able to be with you	0
i love the smell of napalm in the morning	**i love the smell of napalm in the morning**	1
i love the smell of napalm in the morning	**they love the smell of racism in the morning**	1
i love the smell of napalm in the morning	i love the smell of christmas	0

Table 10: Some entries of the CATCHPHRASE dataset. We highlight in **bold** the snowclones in each sentence. A label of 1 indicates that the snowclone is seen in the sentence, while a label of 0 indicates that the snowclone is not present in it.

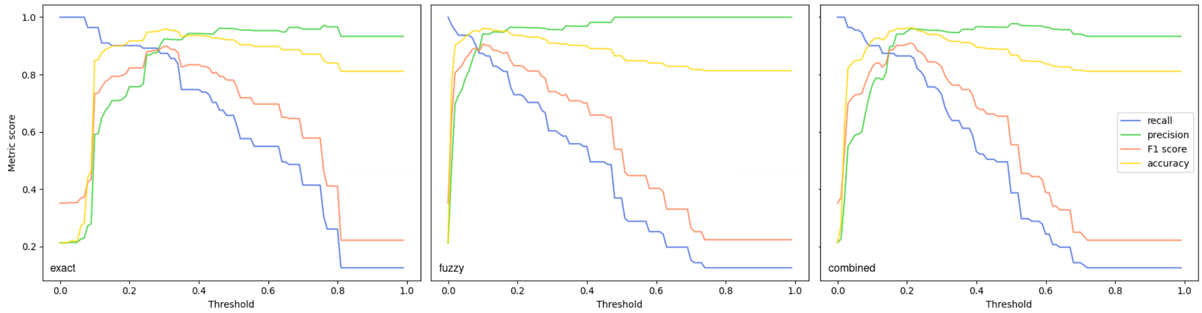

Figure 3: Impact of threshold on recall, precision, F-score and accuracy for the best run on the test partition of CATCHPHRASE with each approach.

B.2 Run parameters

The tested parameters include those of the vectorizer and the threshold at which we consider a candidate to correspond to a snowclone. The possible parameters were as follows:

- ngram: 1,2 | 1,3 | 2,3 | 2,4 | 3,4 | 3,5 | 4,5 | 4,6;
- analyzer: word | char | char_wb;
- threshold: 1, 0.9, 0.8, ... 0.2, 0.1, 0.

The best runs on the test partition of the CATCHPHRASE dataset were the following:

- ASMR$_{exact}$: ngram = 3,4 | analyzer = char | threshold = 0.3;
- ASMR$_{fuzzy}$: ngram = 2,4 | analyzer = word | threshold = 0.3;
- ASMR$_{combined}$: ngram = 1,2 | analyzer = word | threshold = 0.2.

B.3 Resulting ranking

We report some ranked candidates for the snowclone "may the force be with you" in Table 11. For each approach, we use the best parameters found on the train and dev set for the vectorizer with this approach, which is why some candidate's scores may vary. We also report the impact of threshold on the best run with each approach in Figure 3.

B.4 Impact of parameters

To go further, we propose to study, for all partitions and all approaches, the impact of the analyzer chosen in Figure 4 as well as the impact of the n-gram sizes chosen in Figure 5. Both appear to have a limited impact on the obtained F-scores, which may explain the observed robustness of ASMR.

Candidate	Score	Freq
ASMR$_{exact}$		
may the force be with you	1.00	51
may the force be with	0.81	3
the force be with you	0.74	6
the force be with	0.58	1
may the force be you	0.50	1
force be with you	0.44	3
ASMR$_{fuzzy}$		
may the force be with you	1.00	51
may the force be with your	0.69	1
let the force be with you	0.51	6
may the force be good to you	0.29	1
may some of the force be with you	0.22	1
may the gravity force be with you	0.20	3
ASMR$_{combined}$		
may the force be with you	1.00	51
may the force be with your	0.81	1
may some force be with you	0.23	2
may the peace be with you	0.15	1
may the god be with you	0.14	3
may the boop be with you	0.14	1

Table 11: Some candidates obtained with each approach of ASMR.

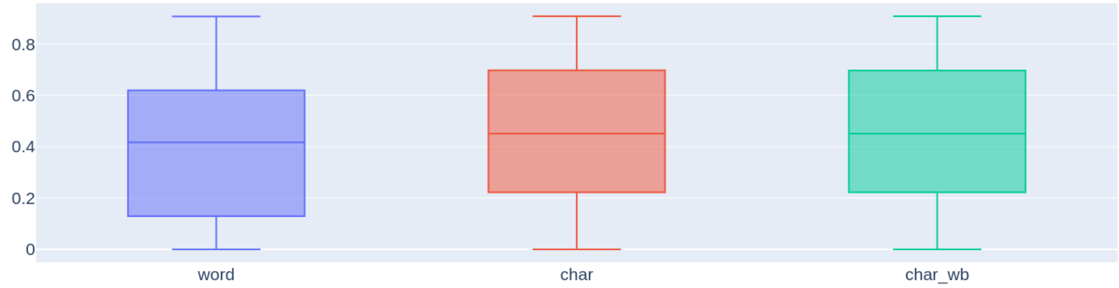

Figure 4: Boxplots of F-scores obtained on all partitions of the CATCHPHRASE dataset with all approaches to ASMR for each analyzer used.

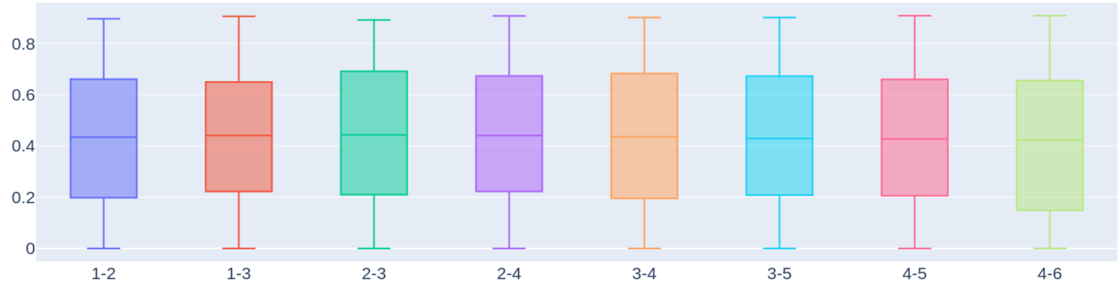

Figure 5: Boxplots of F-scores obtained on all partitions of the CATCHPHRASE dataset with all approaches to ASMR for each ngram sets used.

B.5 Cosine similarity vs Levensthein Distance

To determine that cosine similarity was better suited for our use case than Levenshtein distance, we considered several key factors. First, the time complexity of Levenshtein distance is O^{nm}, whereas cosine similarity operates at a much lower complexity of O^{n}. This makes cosine similarity significantly faster in most scenarios, including ours, where we compute a vector matrix using tf-idf. Additionally, during testing, we observed that Levenshtein distance tends to favor shorter sequences more heavily than cosine similarity, probably resulting in a higher rate of false positives among the top N candidates. Table 12 shows the results we obtained while perfoming the same experiment that the one in Section 4 with Levensthein distance instead of cosine simlarity. We used the PYTHON-LEVENSHTEIN package, which offers a fast C-based implementation of the Levenshtein distance. We observe that the results are consistently lower than those obtained using cosine similarity. Upon reviewing the rankings for several snowclones, we found further evidence supporting our initial assumption: Levenshtein distance tends to favor shorter sequences. This suggests a potential bias that can lead to false positives, which can be described as sequences that share some tokens with the target snowclones but are missing key elements. While this implementation of Levenshtein distance can be used as an alternative to cosine similarity, it introduces a bias toward shorter candidates.

C MWE identification task details

C.1 PARSEME 1.3 metadata

Figure 6 show the number of sentences and MWE for each language in the PARSEME 1.3 corpus. Some languages are much more represented than others. This is especially the case for Portuguese (PT), Romanian (RO), Chinese (ZH) and Czech (CS), which all contain more than 30,000 sentences. Table 13 contains the number of sentences, MWEs and tokens for each partition for each language.

	Recall	Precision	F-score	Accuracy
ASMR$_{exact}$	**86±09**	75±20	77±08	88±06
ASMR$_{fuzzy}$	81±02	79±13	79±03	**91±03**
ASMR$_{combined}$	77±05	76±16	75±06	89±04
SVM (Sweed and Shahaf, 2021)	78±12	**84±13**	**81±NA**	85±08
RoBERTa (Sweed and Shahaf, 2021)	74±18	70±15	72±NA	81±94

Table 12: Results of ASMR for snowclone detection on the CATCHPHRASE test set with a Levensthein distance. For the results of our approaches, the standard deviation is computed on 20 runs.

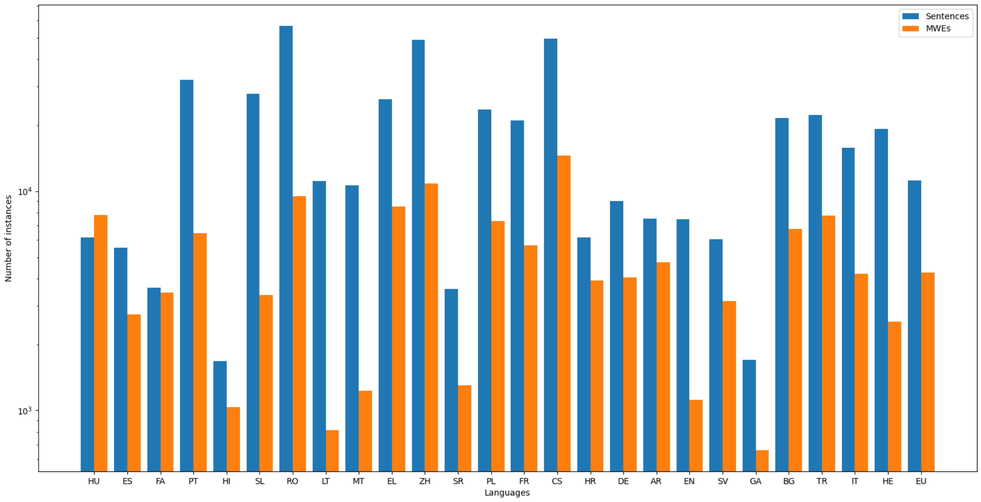

Figure 6: Number of sentences and MWEs for each language in the PARSEME 1.3 corpus.

C.2 Run parameters

The tested parameters all correspond to a threshold for each linguistic information layer we used during our experiments on the PARSEME 1.3 corpus (token level, morphosyntactic level, lemmas and a semantic similarity score). The possible thresholds were 0.1, 0.4, 0.7 and 1. We limited them in order to avoid overloading our calculation server with longer runs. We report in Table 14 the best parameters for each language and for each approach to ASMR. For the semantic scores, we used the PARAPHRASE-XLM-R-MULTILINGUAL-V1 model from the SENTENCE-TRANSFORMERS python package. This model covers all of the 26 languages of ASMR.

C.3 Resulting ranking

For 21 language, we show the top 3 candidates of our ranking system obtained with ASMR$_{combined}$ for a random MWE in Table 15.

D Error analysis details

We show the the F-scores obtained for each subset of MWE for each language in the PARSEME 1.3 corpus for each approach in Table 16, Table 17 and Table 18.

	# Sent.			# MWE			# Tokens		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
AR	6,091	342	1,050	3,841	228	680	252,515	14,751	44,551
BG	15,950	1,380	4,269	4,969	431	1,304	353,748	30,980	95,685
CS	42,288	1,725	5,418	12,405	523	1,608	711,213	28,697	93,283
DE	6,475	628	1,893	2,912	281	848	125,081	12,046	36,434
EL	21,983	1,077	3,115	7,128	348	1,032	587,001	28,833	82,590
EN	2,150	1,302	3,984	317	199	598	35,534	21,660	67,009
ES	3,424	521	1,570	1,732	256	751	112,906	17,333	52,125
EU	5,033	1,441	4,684	1,932	560	1,754	70,017	20,957	66,833
FA	2,364	321	932	2,249	303	901	40,110	5,430	16,028
FR	14,540	1,580	4,841	3,921	437	1,297	364,414	40,107	121,321
GA	330	318	1,057	127	134	398	7,104	7,680	24,123
HE	14,035	1,296	3,869	1,855	171	507	283,984	26,766	77,731
HI	399	322	963	242	200	592	8,641	6,786	20,003
HR	3,357	672	2,104	2,131	439	1,332	77,599	15,329	50,018
HU	2,139	1,000	3,020	2,664	1,259	3,837	54,658	25,205	76,473
IT	10,641	1,202	3,885	2,854	324	1,032	292,065	32,652	106,072
LT	2,281	2,181	6,642	163	161	488	42,782	41,421	124,309
MT	6,460	975	3165	749	119	358	154,979	22,924	74,382
PL	18,037	1,421	4,089	5,595	430	1,288	303,628	23,865	68,647
PT	24,594	1,867	5,601	4,926	375	1,125	557,486	42,855	127,728
RO	26,889	7,668	22,107	4,562	1,257	3,689	479,681	139,314	395,913
SL	15,220	3,054	9,551	1,834	376	1,153	321,377	64,429	200,381
SR	1,382	544	1,660	492	203	609	33,839	13,558	39,970
SV	2,795	765	2,466	1,467	421	1,269	44,917	12,335	39,607
TR	16,730	1,396	4,180	5,824	466	1,439	248,697	20,679	62,793
ZH	44,103	1,215	3,611	9,744	274	801	738,713	19,936	61,698

Table 13: Number of sentences, MWEs and tokens for each language and for each partition in the PARSEME 1.3 corpus.

	ASMR_{exact}				ASMR_{fuzzy}				$\mathrm{ASMR}_{combined}$			
	tok	upos	lem	sem	tok	upos	lem	sem	tok	upos	lem	sem
AR	0.1	0.4	0.7	0.1	0.1	0.4	0.7	0.1	0.1	0.7	0.1	0.1
BG	0.4	0.4	0.7	0.1	0.4	0.4	0.7	0.1	0.4	0.4	0.7	0.1
CS	0.4	0.4	0.7	0.4	0.4	0.4	0.7	0.4	0.4	0.4	0.7	0.1
DE	0.1	0.7	0.7	0.4	0.1	0.4	0.7	0.4	0.1	0.7	0.4	0.4
EL	0.1	0.7	0.7	0.1	0.1	0.7	0.4	0.1	0.1	0.7	0.7	0.1
EN	0.1	0.7	0.7	0.4	0.1	0.7	0.1	0.4	0.1	0.7	0.7	0.4
ES	0.1	0.1	0.7	0.4	0.1	0.1	0.7	0.4	0.1	0.1	0.7	0.4
EU	0.1	0.7	0.7	0.4	0.1	0.7	0.4	0.4	0.1	0.7	0.4	0.4
FA	0.1	0.7	0.4	0.1	0.1	0.7	0.7	0.1	0.1	0.7	0.7	0.4
FR	0.1	0.7	0.7	0.1	0.1	0.7	0.4	0.1	0.1	0.7	0.4	0.1
GA	0.1	0.4	0.7	0.4	0.1	0.7	0.1	0.4	0.1	0.7	0.7	0.4
HE	0.4	0.1	0.7	0.4	0.4	0.1	0.7	0.4	0.4	0.1	0.7	0.4
HI	0.1	0.7	0.7	0.1	0.1	0.7	0.7	0.4	0.1	0.7	0.1	0.1
HR	0.1	0.1	0.7	0.1	0.1	0.1	0.7	0.1	0.1	0.7	0.4	0.1
HU	0.1	0.1	0.4	0.7	0.1	0.1	0.4	0.7	0.1	0.1	0.4	0.7
IT	0.1	0.7	0.7	0.4	0.1	0.4	0.7	0.7	0.1	0.7	0.4	0.4
LT	0.1	0.7	0.7	0.4	0.1	0.7	0.4	0.4	0.1	0.7	0.7	0.4
MT	0.1	0.7	0.7	0.4	0.1	0.7	0.4	0.4	0.1	0.7	0.7	0.4
PL	0.1	0.1	0.7	0.4	0.1	0.1	0.7	0.4	0.1	0.1	0.7	0.1
PT	0.1	0.7	0.7	0.7	0.1	0.7	0.7	0.4	0.1	0.7	0.7	0.7
RO	0.1	0.7	0.7	0.4	0.1	0.7	0.4	0.4	0.1	0.7	0.4	0.4
SL	0.1	0.7	0.7	0.1	0.1	0.7	0.4	0.1	0.1	0.7	0.4	0.1
SR	0.1	0.1	0.7	0.1	0.1	0.1	0.7	0.1	0.1	0.1	0.7	0.1
SV	0.1	0.4	0.7	0.4	0.1	0.4	0.7	0.4	0.1	0.7	0.4	0.4
TR	0.4	0.1	0.7	0.1	0.4	0.1	0.7	0.1	0.4	0.1	0.7	0.1
ZH	0.7	0.7	0.7	0.7	0.7	0.7	0.1	0.1	0.7	0.4	0.7	0.7

Table 14: Best run parameters for each language for each approach for each linguistic information layer: token (tok), morphosyntactic (upos), lemmas (lem) and for the semantic similarity (sem).

Language	Candidate	mean	tok	upos	lem	sem
BG	решаване на проблеми	0.99	0.99	1.0	1.0	0.96
	решаване на проблема	0.98	0.93	1.0	1.0	0.99
	решаване на проблемите	0.97	0.91	1.0	1.0	0.95
CS	mít problém	1.0	1.0	1.0	1.0	1.0
	mít problémů	0.97	0.91	1.0	1.0	0.98
	má problém	0.91	0.69	1.0	1.0	0.95
DE	der entscheiden	1.0	1.0	1.0	1.0	1.0
	Der entscheiden	0.97	1.0	1.0	1.0	0.87
	den entscheiden	0.97	0.89	1.0	1.0	0.98
EL	το παίρνει	0.92	0.83	1.0	1.0	0.85
	Το παίρνει	0.9	0.83	1.0	1.0	0.76
	Ο παίρνει	0.9	0.84	1.0	1.0	0.76
EN	Look forward	1.0	1.0	1.0	1.0	0.99
	look forward	1.0	1.0	1.0	1.0	1.0
	looking forward	0.91	0.7	1.0	1.0	0.94
ES	informar de	1.0	1.0	1.0	1.0	1.0
	informa de	0.93	0.81	1.0	1.0	0.93
	informaron de	0.92	0.81	1.0	1.0	0.88
EU	aintzat hartu	1.0	1.0	1.0	1.0	1.0
	aintzat har	0.99	0.98	1.0	1.0	0.97
	aintzat hartuz	0.95	0.88	1.0	1.0	0.93
FR	se rendre compte	1.0	1.0	1.0	1.0	1.0
	s’ rendre compte	0.95	0.8	1.0	1.0	0.99
	se rendant compte	0.88	0.55	1.0	1.0	0.96
GA	baint le	1.0	1.0	1.0	1.0	1.0
	baint leis	0.96	0.94	1.0	1.0	0.92
	bhaint leo	0.87	0.67	1.0	1.0	0.79
HR	nastaviti s	0.99	0.98	1.0	1.0	0.99
	Nastaviti s	0.98	0.98	1.0	1.0	0.95
	nastavi s	0.92	0.71	1.0	1.0	0.98
HU	kötött szerződés	1.0	1.0	1.0	1.0	1.0
	kötött szerződést	0.98	0.94	1.0	1.0	0.99
	kötött szerződésben	0.97	0.91	1.0	1.0	0.98
IT	si prestare	1.0	1.0	1.0	1.0	1.0
	Si prestare	0.98	1.0	1.0	1.0	0.9
	Si prestata	0.92	0.74	1.0	1.0	0.92
LT	sprendimas priimtas	0.92	0.81	1.0	1.0	0.89
	Sprendimas priimtas	0.91	0.81	1.0	1.0	0.84
	sprendimą priimti	0.91	0.65	1.0	1.0	0.98
MT	Il- industrija	1.0	1.0	1.0	1.0	0.98
	l- industrija	0.98	0.99	1.0	1.0	0.94
	Iż- industrija	0.96	0.91	1.0	1.0	0.94
PL	spodziewać się	1.0	1.0	1.0	1.0	1.0
	spodziewają się	0.89	0.69	1.0	1.0	0.88
	spodziewał się	0.89	0.77	1.0	1.0	0.78
PT	ter qualidade	1.0	1.0	1.0	1.0	1.0
	tem qualidade	0.94	0.83	1.0	1.0	0.94
	teve qualidade	0.91	0.75	1.0	1.0	0.9
RO	beneficia de	1.0	1.0	1.0	1.0	1.0
	beneficiat de	0.96	0.87	1.0	1.0	0.97
	beneficiază de	0.94	0.8	1.0	1.0	0.97
SL	se privoščiti	1.0	1.0	1.0	1.0	1.0
	se privoščite	0.97	0.9	1.0	1.0	0.98
	si privoščiti	0.97	0.93	1.0	1.0	0.95
SR	biti u problema	0.98	0.94	1.0	1.0	1.0
	je u problem	0.93	0.79	1.0	1.0	0.93
	sam od problem	0.77	0.5	1.0	0.73	0.83
SV	ta reda på	1.0	1.0	1.0	1.0	1.0
	Ta reda på	0.99	1.0	1.0	1.0	0.97
	får reda på	0.81	0.6	1.0	0.65	0.97
TR	teşekkür etti	0.98	0.97	1.0	1.0	0.97
	teşekkür ederim	0.97	0.89	1.0	1.0	1.0
	teşekkür eden	0.97	0.95	1.0	1.0	0.94

Table 15: top 3 results obtained for a random MWE for 21 languages in PARSEME 1.3 with ASMR$_{combined}$.

	Tok-based	Continuous	Discontinuous	Seen	Unseen	Variant	Identical
AR	43.6	48.6	26.5	55.8	00.7	46.7	69.6
BG	62.9	66.7	47.0	70.0	00.0	51.1	80.4
CS	67.6	73.6	49.7	69.1	01.7	61.1	86.6
DE	37.8	38.7	22.0	44.1	00.0	38.3	51.8
EL	59.8	63.6	50.5	72.1	00.0	62.2	85.6
EN	55.5	62.1	47.1	83.2	00.0	73.1	93.3
ES	56.1	59.6	38.2	68.8	00.0	57.8	84.8
EU	75.6	83.7	45.8	81.6	00.0	69.0	95.8
FA	71.3	75.5	39.4	84.5	00.8	75.8	93.2
FR	71.2	75.2	61.0	80.2	00.0	73.1	86.6
GA	30.7	40.4	16.3	62.7	00.0	53.9	92.9
HE	46.4	48.0	38.5	74.2	00.0	54.8	92.1
HI	59.8	62.1	23.9	87.3	00.0	77.4	96.3
HR	75.1	83.3	62.9	87.1	00.0	74.9	93.3
HU	43.0	26.1	62.8	33.1	00.0	48.5	28.9
IT	61.5	67.0	47.8	76.9	00.0	68.1	88.9
LT	38.7	41.1	41.4	76.8	00.0	72.6	98.3
MT	19.5	18.0	11.3	32.1	00.0	31.8	32.5
PL	73.5	80.6	54.4	85.5	00.0	79.5	92.3
PT	59.0	61.3	55.2	82.8	20.3	79.4	90.0
RO	73.8	76.4	63.9	74.9	00.0	46.7	87.9
SL	40.2	43.3	37.4	44.8	00.0	40.4	58.1
SR	53.4	56.4	44.6	80.5	00.0	75.0	94.1
SV	54.1	39.2	56.1	51.9	00.0	58.5	45.9
TR	64.7	69.5	13.5	71.1	00.7	64.5	84.3
ZH	35.5	27.4	32.8	37.6	00.0	31.8	38.7
Mean	55.0	57.1	41.9	68.0	00.8	60.1	78.6

Table 16: F-score obtained for each subset of MWE in each language with the PARSEME 1.3 corpus, using ASMR$_{exact}$.

	Tok-based	Continuous	Discontinuous	Seen	Unseen	Variant	Identical
AR	34.6	38.5	11.0	45.3	05.8	33.9	60.6
BG	58.5	63.7	21.0	67.2	06.3	42.0	77.9
CS	56.0	66.4	20.5	60.6	05.5	46.6	84.7
DE	32.6	29.5	06.2	36.5	01.7	28.1	46.3
EL	51.9	60.5	22.4	61.5	08.7	50.2	75.0
EN	41.8	54.6	10.8	69.3	03.8	56.4	80.2
ES	49.8	53.5	17.0	65.2	02.6	51.7	81.5
EU	68.8	76.2	17.1	78.1	03.3	62.8	92.2
FA	72.3	77.1	13.1	84.6	25.8	75.4	92.6
FR	57.8	63.9	22.9	68.4	02.8	55.1	78.1
GA	26.8	36.7	04.8	57.5	06.3	48.1	80.8
HE	43.1	45.7	16.1	69.6	06.9	46.5	88.6
HI	61.1	62.6	05.3	90.8	14.0	84.8	95.8
HR	61.8	73.5	25.9	73.9	03.6	62.1	79.5
HU	39.5	25.5	17.9	28.4	05.0	39.5	25.6
IT	54.6	59.6	17.9	72.1	04.1	61.4	84.8
LT	29.5	39.3	11.7	61.0	02.4	54.3	91.8
MT	18.8	17.5	04.9	30.4	05.7	29.9	31.0
PL	63.1	72.5	25.8	76.5	04.1	66.2	86.8
PT	42.1	52.5	00.0	60.3	20.7	49.2	78.4
RO	64.1	66.9	26.3	68.2	01.9	43.2	77.7
SL	31.0	37.4	15.4	36.4	01.3	29.7	51.9
SR	48.1	54.5	22.3	73.0	06.9	65.7	89.7
SV	47.5	37.8	23.1	44.9	07.1	45.5	43.8
TR	62.2	65.0	05.6	71.6	09.0	66.0	81.8
ZH	35.3	29.9	03.0	34.2	10.5	13.3	36.7
Mean	48.1	52.3	14.8	61.0	06.8	50.3	72.8

Table 17: F-score obtained for each subset of MWE in each language with the PARSEME 1.3 corpus, using ASMR$_{fuzzy}$.

	Tok-based	Continuous	Discontinuous	Seen	Unseen	Variant	Identical
AR	40.2	44.4	21.0	54.1	06.2	46.2	66.1
BG	59.5	64.7	38.4	67.6	06.6	45.7	78.1
CS	64.8	70.8	47.3	68.3	04.9	60.0	86.8
DE	35.3	29.4	20.5	39.7	02.8	39.1	40.3
EL	59.1	64.8	48.5	70.0	10.0	62.2	80.7
EN	52.9	58.6	45.3	81.6	00.8	72.2	91.2
ES	52.4	56.0	35.5	67.6	01.3	56.8	82.6
EU	71.5	79.6	41.8	81.0	03.4	68.7	94.8
FA	73.2	78.1	39.4	84.8	23.0	76.4	92.9
FR	67.2	72.0	55.6	80.5	01.0	73.8	86.4
GA	31.7	40.2	16.2	69.0	02.3	61.1	91.2
HE	45.3	47.1	34.9	72.8	05.0	53.1	90.9
HI	58.5	61.7	19.0	86.9	04.1	77.1	95.8
HR	70.0	77.2	57.4	84.8	03.5	73.4	90.7
HU	42.6	25.8	58.5	32.6	03.6	48.1	28.4
IT	60.1	66.5	44.5	77.4	02.1	69.1	88.6
LT	38.5	40.3	41.6	75.6	03.0	71.6	96.2
MT	15.2	13.1	09.4	25.0	01.6	24.7	25.4
PL	67.8	75.0	47.5	83.3	04.2	76.8	90.8
PT	55.9	60.4	48.3	81.3	23.2	77.6	88.9
RO	67.6	71.1	56.4	73.6	02.6	47.5	85.2
SL	37.9	40.2	35.0	44.3	01.7	40.2	56.8
SR	52.9	56.3	42.1	79.8	04.6	74.1	93.8
SV	52.3	37.8	52.6	50.6	02.7	56.9	45.0
TR	59.4	65.0	08.6	68.1	06.2	61.8	80.4
ZH	36.5	28.4	32.9	37.8	05.0	30.5	39.2
Mean	52.6	54.8	38.4	66.9	05.2	59.4	76.4

Table 18: F-score obtained for each subset of MWE in each language with the PARSEME 1.3 corpus, using ASMR$_{combined}$.