Flexible Lexicalization in Rule-based Text Realization

Avril Gazeau, François Lareau
OLST, Université de Montréal
Montréal, Canada
first.lastname@umontreal.ca

Abstract

GenDR is a text realizer that takes as input a graph-based semantic representation and outputs the corresponding syntactic dependency trees. One of the tasks in this transduction is lexicalization, i.e., choosing the right lexical units to express a given semanteme. To do so, GenDR uses a semantic dictionary that maps semantemes to corresponding lexical units in a given language. This study aims to develop a flexible lexicalization module to automatically build a rich semantic dictionary for French. To achieve this, we tried two methods. The first one consisted in extracting information from the *French Lexical Network*, a large-scale French lexical resource, and adapting it to GenDR. The second one was to test a contextual neural language model's ability to generate potential additional lexicalizations. The first method significantly broadened the coverage of GenDR, while the additional lexicalizations produced by the language model turned out to be of limited use, which brings us to the conclusion that it is not suited to perform the task we've asked from it.

Keywords: text realization, semantics-syntax interface, lexicalization, language models

1. Introduction

GenDR[1] (Lareau et al., 2018) is a generic, multilingual symbolic linguistic realizer that runs on a graph transducer called MATE (Bohnet et al., 2000; Bohnet and Wanner, 2010). It is based on the Meaning-Text Theory (Žolkovskij and Mel'čuk, 1967; Kahane, 2003; Milićević, 2006; Mel'čuk, 2016). Its input is an abstract semantic representation (SemR) (Mel'čuk, 2012) and its output is a set of corresponding dependency trees (Mel'čuk, 1988). Through a set of rules and dictionaries, it builds, from a SemR, a set of deep-syntactic representations (DSyntRs) and then a set of surface-syntactic representations (SSyntRs). Figure 1 shows such representations for a simple sentence. One of the key tasks in this transduction is lexicalization, i.e., choosing the right lexical units (LUs) to express the semantemes from the input in a given language. To do so, it relies on a semantic dictionary (SD) that maps semantemes to their corresponding LUs, regardless of their part of speech (POS) or diathesis. For instance, 'cause' would be mapped to *cause*$_{\text{V}}$, *cause*$_{\text{N}}$, *due*, *because*, *consequence*, *reason*$_{\text{N}}$, and so on.

The richer the SD for a given language, the more SSyntRs GenDR can generate, allowing for more varied and more fluid realizations. Basic SDs have been put together for English (Galarreta-Piquette, 2018), Chinese (Zhao, 2018), French (Lareau et al., 2018), Lithuanian (Dubinskaitė, 2017) and Persian (Lareau et al., 2018), but all of them have two major limitations:

1. They are very small, and broadening their coverage manually would be costly.

[1]The name stands for *generic deep realizer*.

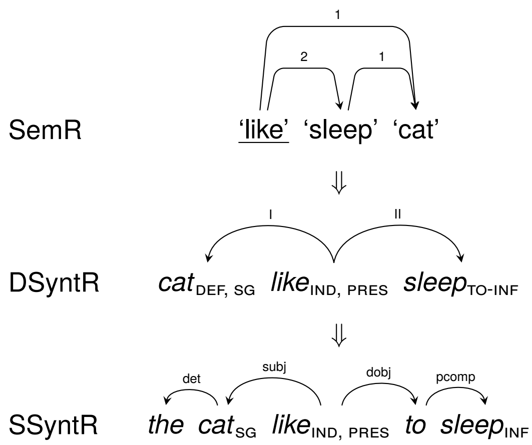


Figure 1: Three levels of representation for the sentence *The cat likes to sleep*

2. They contain errors and inconsistencies, because they were compiled manually by different people without clear guidelines.

In this paper, we tried two methods to resolve these issues:

1. Deriving a SD automatically from a large existing lexical resource, namely French Lexical Network (LN-fr).
2. Harnessing neural language models to further increase the SD's coverage.

The following sections describe our experiments with these methods.

LREC-COLING 2024, pages 7366–7376

2. SD based on LN-fr

2.1. French Lexical Network (LN-fr)

LN-fr (Polguère, 2009, 2014; Ollinger and Polguère, 2020) is a lexical resource based on the Explanatory Combinatorial Lexicology (ECL) framework (Mel'čuk et al., 1995; Mel'čuk, 1995; Apresjan, 2000). It is formally a small-world graph (Watts and Strogatz, 1998) where most of the nodes, and the ones that are relevant for our purposes, are LUs, each associated with a specific meaning (as such, it is disambiguated) and a citation form, that we call its *normalized name*.

Most of the edges, and the ones that are relevant to us, are lexical functions (LFs).[2] LFs (Mel'čuk, 1996; Mel'čuk and Polguère, 2021) represent common relations between LUs, either paradigmatic (synonymy, nominalization, verbalization, etc.) or syntagmatic (collocational intensifiers, light verbs, etc.). Formally, a LF is a function that takes as argument a LU and returns a set of LUs that manifest a specific semantico-syntactic pattern that is recurrent within and across languages, such as:

- Intensification:
  $\texttt{Magn}$(*change*$_{\text{N}}$)={*radical*, *drastic*, *sea*}
- Derivative noun for agent, patient, etc.:
  $\texttt{S}_1$(*work*) = {*worker*}
  $\texttt{S}_2$(*drink*$_{\text{V}}$) = {*drink*$_{\text{N}}$, *beverage*}
- Support verb:
  $\texttt{Oper}_1$(*nap*$_{\text{N}}$) = {*take*, *have* [a ∼]}
  $\texttt{Func}_0$(*event* = {*happen*, ⌜*take place*⌝}
- Derivative modifier:
  $\texttt{Adv}_1$(*affection*) = {*affectionately, with* ∼}
  $\texttt{A}_2$(*control*$_{\text{N}}$) = {*under* [∼]}
- Etc.

Figure 2 shows a sample of LN-fr's graph.

2.2. Methodology

Since each node is associated with only one semanteme, its normalized name can be used to label that semanteme. Hence, we used each node's normalized name as a SD entry, and mapped it to a set containing at first only its trivial lexicalization–a copy of the semanteme's normalized name, since a semanteme can always be expressed as its label. Then, we added to this set the values of semantically empty paradigmatic lexical functions (SEPLFs) applied to this LU. A paradigmatic LF[3] is a LF that encodes a derivational semantic relation (Mel'čuk and Polguère, 2021). A subset of paradigmatic LFs are semantically empty: they encode a very special type of derivation, a substitution relation. These are what we call SEPLFs and can be thought of as a kind of synonymy relation, but without limitations regarding the part of speech. For example, the relations between *cause*$_{\text{V}}$, *cause*$_{\text{N}}$, *because* or *due* are SEPLFs. A SEPLF can be exact (e.g., *agree–agreement*) or approximate (e.g., *actor–actress*, or *run*$_{\text{V}}$*–sprint*$_{\text{V}}$).

In order to retrieve from LN-fr all and only the LFs that are SEPLFs, we created a list of regular expressions that match all possible SEPLFs and annotated them as either exact (0) or approximate (1). We also marked them as POS-preserving or not. Table 1 shows these regular expressions and table 2 shows the SEPLFs we matched in the LN-fr; the list of all 64 can be found in the appendix (5).

The number(s) following the SEPLF refer to the syntactic actant(s) of the argument the function applies to. Some examples of the SEPLFs shown in table 2 encode a:

- Derivative verb:
  $\texttt{V}_0$(*access*$_{\text{N}}$) = {*access*$_{\text{V}}$}
- Derivative noun:
  $\texttt{S}_0$(*involve*) = {*involvement*}
- Syntactic actant permutation:
  $\texttt{Conv}_{21}$(*precede*) = {*follow*}
- Generic term:
  $\texttt{Gener}$(*sardine*) = {*fish*}

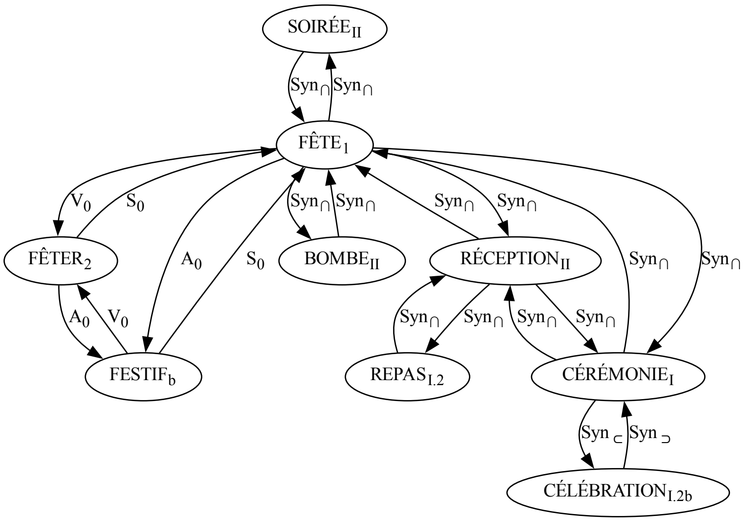
After adding the lexicalizations found through SEPLFs, we applied recursively this SEPLF-based search, limiting it with a maximum approximation parameter (MAP). The MAP is an integer that represents the maximum number of meaning approximations allowed in the search. As we explore the graph of SEPLFs, we add the approximation value associated with each LF, and when a path yields a sum of its approximation scores over the value of MAP, we stop exploring it further and we backtrack to explore other paths, until we have explored all paths with an approximation sum within the desired MAP. The LUs visited through this exploration are all added to the lexicalization set of the initial entry. For instance, for the graph in figure 3, with $MAP = 1$, the lexicalization set for A would be $\{A, B, C, D\}$. Thus, by adjusting the value of MAP, one can derive a more or less flexible semantic dictionary.

2.3. Results and evaluation

With our flexible lexicalization module, we built a SD compatible with GenDR containing 29,399 entries (the number of nodes in the LN-fr) mapped to the same number of unique LUs. These LUs are linked to each other through a minimum of 49,235 lexicalization links when $MAP = 0$, and through 572,686

[2] For other types of nodes and edges, see Polguère (2014). For the sake of simplicity, in this paper we talk about LN-fr's nodes as if they were all LUs, and about its edges as if they were all LFs.

[3] In some cases, a paradigmatic LF can have a syntagmatic output and a syntagmatic LF can have a paradigmatic output (Mel'čuk and Polguère, 2021).



Glosses	
soirée$_{II}$	'soirée'
fête$_1$	'party$_N$'
fêter$_2$	'party$_V$'
festif$_b$	'festive'
bombe$_{II}$	'bash'$_N$
réception$_{II}$	'function'$_N$
repas$_{I.2}$	'meal'
cérémonie$_I$	'ceremony'
célébration$_{1.2b}$	'ceremony'

Figure 2: Semantically empty or near-empty derivations for *fête*$_1$ ('party$_N$') in LN-fr

Regex	Approx.	Same POS
`A_\d`	0	F
`A_\d[∩ ⊂⊃]`	1	F
`A_\dPred`	0	F
`Adv_\d`	0	F
`Adv_0[∩ ⊂⊃]`	1	F
`Conv_\d+`	0	T
`Conv_\d+Pred`	0	F
`Conv_d+[∩ ⊂⊃]`	1	T
`Figur`	0	F
`Gener`	1	F
`Pred`	0	F
`S_0[∩ ⊂⊃]`	1	F
`S_0\usual`	1	F
`S_0`	0	F
`S_0A_\d`	0	F
`S_0Conv_\d+`	0	F
`S_0Pred_[∩ ⊂⊃]`	1	F
`S_0Pred`	0	F
`Syn_[∩ ⊂⊃]\.+`	1	T
`Syn_[∩ ⊂⊃]`	1	T
`Syn`	0	T
`V_0[∩ ⊂⊃]`	1	F
`V_0`	0	F
`V_0Conv_\d+`	0	F

Table 1: SEPLF patterns

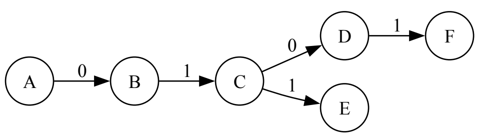


Figure 3: Approximate synonymy

Name	Approx.	Same POS
`A_1`	0	F
`A_1Pred`	0	F
`A_2`	0	F
`Adv_1`	0	F
`Adv_2`	0	F
`Conv_21`	0	T
`Conv_23`	0	T
`Conv_32`	0	T
`Figur`	0	F
`Gener`	1	F
`Pred`	0	F
`S_0_∩`	1	F
`S_0_⊂`	1	F
`S_0_⊃`	1	F
`S_0`	0	F
`S_0A_1`	0	F
`S_0Conv_21`	0	F
`Syn_∩`	1	T
`Syn_⊂`	1	T
`Syn_⊂^sex`	1	T
`Syn_⊃`	1	T
`Syn_⊃^sex`	1	T
`Syn`	0	T
`V_0`	0	F

Table 2: Some of the SEPLFs extracted from LN-fr

links when $MAP = 5$. Thus, GenDR's french SD has been considerably broadened, since it initially contained around 1,425 entries mapped to around 1,555 lexicalizations. GenDR's paraphrasing capacity has been significantly heightened. Below are a few of the sentences we were able to generate for the meaning 'sleep(cat)' using a SD generated with $MAP = 0$.

(1) *Le chat dort.*
'The cat is sleeping.'

(2) *Le chat en écrase.*
'The cat is sleeping like a log.'

(3) *Le chat pieute.*
'The cat is sleeping.'

(4) *Le chat roupille.*
'The cat is snoozing.'

(5) *Le chat sommeille.*
'The cat is slumbering.'

(6) *Le chat est endormi.*
'The cat is asleep.'

(7) *Le chat fait un roupillon.*
'The cat is taking a snooze.'

(8) *Le chat fait un somme.*
'The cat is taking a nap.'

Our approach could be used as-is to enhance SDs for other languages, as long as there is a similar LN for it. This highlights the need for rich lexical resources such as LN-fr.

3. Expanding the SD with CamemBERT

3.1. CamemBERT

CamemBERT (Martin et al., 2020) is a French contextual word embedding model of the Transformers kind (Vaswani et al., 2017) based on BERT (Devlin et al., 2019), itself based on the distributional hypothesis (Harris, 1954; Firth, 1957), which states that words found in similar contexts have similar meanings. CamemBERT can weight the context tokens of a specific token through an attention mechanism (Vaswani et al., 2017) in order to calculate its position in the vector space. Since each vector is dependant on the token's context, it can be considered semantically disambiguated and mapped to unique coordinates. Since only one meaning is associated to each LU or semanteme—the objects making up the SD—we chose to test our method with CamemBERT because of this disambiguation property.

To learn to calculate vectors in their context, CamemBERT is trained to attain a Masked Language Modeling (MLM) objective and on a task of Next Sentence Prediction (NSP) (Devlin et al., 2019), two properties we harness in this study.

3.2. Methodology

To broaden the SD, our objective was to calculate the vectors corresponding to the entries' label and ask CamemBERT for their closest neighbours which, according to the distributional hypothesis, should have a similar meaning and thus should be appropriate lexicalizations.

Listing 1: Inputs for basic and `<SEP>` methods

```
The dog <MASK> after the postman.

The dog runs after the postman. <SEP>
    The dog <MASK> after the postman.
```

In LN-fr, each node is associated with one or more sentences putting it in context, and these sentences have been annotated by the lexicographers to explicitly mark the character offset corresponding to the LU, as in table 3. Hence, we were able to mask the part of the sentences corresponding to the nodes sharing their normalized name with the SD entries for CamemBERT. We ignored all idioms, because CamemBERT interprets the mask as one token, so it would never assume it has to propose multiple tokens with a meaning similar to an idiom. As such, we were left with 45,134 sentences for 23,277 individual LN-fr nodes, which we masked before asking CamemBERT for the 10 first candidates to replace the mask, yielding 451,340 candidates in total. Each candidate is given a certainty score between 0 and 1, produced by a softmax function. Table 4 shows an example of CamemBERT's output for *run* in a sentence.

We tested two methods to produce the candidates. The first one, the *basic method*, consisted in masking the keyword in the sentence directly and asking CamemBERT for the most likely candidates. The second one, which we call the *<SEP> method*, is inspired from Qiang et al. (2019). It consisted in first showing the full, unmasked sentence to CamemBERT, then using the special token `<SEP>` used in the NSP training to let CamemBERT know another sequence was to follow, and then showing it again the same sentence but with the keyword masked. Both are shown in listing 1. The purpose of the second method was to evaluate if, having first seen the masked token, CamemBERT would stay on track semantically and suggest better candidates than with the basic method, which tends to yield more erratic candidates.

3.3. Results and evaluation

We calculated the precision, the recall and the F-score of the two methods against SDs with a MAP between 0 and 5 (referred to as SD_0 to SD_5). We also normalized the candidates' certainty scores generated for each SD entry by dividing them by the highest score (so that the first candidate has a normalized score of 1). Then, we divided the candidates produced for each entry into subsets according to different normalized score thresholds. We wanted to see if the score was correlated with the usability of the candidate for GenDR's SD. We com-

Node	Sentence	Offset
*manger*I.1a 'eat'	Je [...] **mangeai** un sandwich de pain de mie à la tomate et au thon. 'I [...] **ate** a white-bread tomato and tuna sandwich.'	(42, 49)

Table 3: Sentence associated to the LN-fr node *manger*I.1a

Sequence	Candidate	Score
The dog **runs** after [...]	runs	0.19
	dies	0.076
	passes	0.045
	is	0.043
	arrives	0.039
	...	...

Table 4: Example candidates for *run*

Method	SD_0	SD_1	SD_5
Basic	0.3 %	2.4 %	4.0 %
`<SEP>`	0.6 %	4.4 %	6.5 %

Table 5: Precision micro-average

pared the candidates with the lexicalizations from LN-fr, because it stands to reason that if CamemBERT's output is good enough to reproduce the SDs to a large extent, many generated candidates absent from the SD could enrich an existing entry, and new entries could be lexicalized automatically with CamemBERT. Moreover, since LN-fr was built manually by expert lexicographers, it is a good golden standard to compare CamemBERT to.

3.3.1. Preparation

We first cleaned the data to make the results as neutral and realistic as possible, following these steps:

1. We removed the trivial lexicalizations from both the lexicalizations set and the candidates set for each entry. Indeed, we do not want to know wether CamemBERT is able to find the trivial lexicalization, because we can always reproduce it by just copying an entry's label.
2. We kept only a subset of the lexicalizations for each entry that shared their POS with the entry. Since CamemBERT has a very good sense of syntax, it would be unfair to evaluate its ability to suggest a candidate with a different POS than the mask it was supposed to replace in the LN-fr sentences.
3. We lemmatized the candidates using the French LEFFF Lemmatizer in Python[4], based on Sagot (2010)'s work. This allows us to compare them with the lexicalizations, whose normalized names are in a lemmatized form.
4. We removed the entries mapped to lexicalization sets that remained empty after the removal of the trivial lexicalization. Indeed, we thought it was unfair to ask CamemBERT to generate candidates with a similar meaning to an entry even human lexicographers couldn't find a different lexicalization for.

[4] `https://github.com/ClaudeCoulombe/FrenchLefffLemmatizer`

3.3.2. Precision

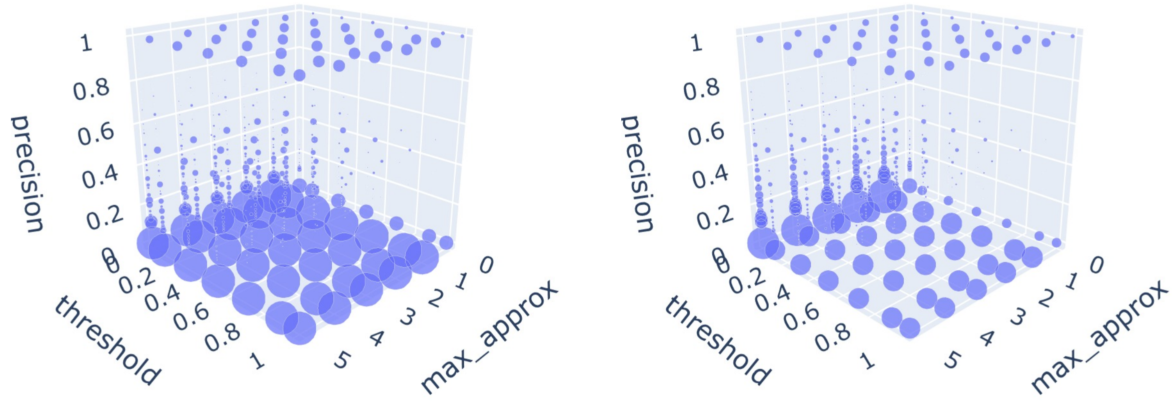
The precision corresponds to the overlap between the lexicalization set produced with LN-fr and the candidate set provided by CamemBERT, divided by the number of candidates produced by CamemBERT for the keyword. We calculated it against SD_0 to SD_5. We first calculated the global precision, or micro-average of the precision, where we calculated the sum of the overlapping candidates for all of the entries and divided it with the total number of generated candidates for a whole SD. Table 5 shows our results. The precision culminates when the candidates are compared with SD_5.

Figures 4 shows the precision we obtained with the basic method and the `<SEP>` method, respectively. We divided the candidate sets in different subsets according to a minimal threshold for the candidates' normalized score. The size of the data points represents the number of candidate sets that have obtained a given precision score according to the normalized score of their candidates and the SD's MAP they were found in.

The majority of the candidate sets obtained a precision score of 0, although a non-negligible portion obtained a precision score of 1. When the minimal certainty threshold is 0 (all candidates are considered), the precision scores are more distributed, instead of being concentrated at 0 or 1. This effect is even more striking with the `<SEP>` method.

3.3.3. Recall

The recall corresponds to the overlap between the two sets divided by the number of elements found in the lexicalizations set. Table 6 shows the recall's micro-average, where we calculated the sum of the overlapping candidates for all of the entries and divided it with the total number of lexicalizations in



(a) Basic method

(b) <SEP> method

Figure 4: Precision

Method	SD$_0$	SD$_1$	SD$_5$
Basic	9.3 %	12.3 %	6.8 %
<SEP>	17.2 %	19.6 %	9.9 %

Table 6: Recall micro-average

Method	SD$_0$	SD$_1$	SD$_5$
Basic	0.51 %	4.10 %	5.00 %
<SEP>	1.10 %	7.15 %	7.80 %

Table 7: F-score micro-average

a SD. It culminates when calculated against the SD$_1$ and progressively decreases with the more approximate SDs.

Figure 5 shows the recall we obtained for every candidates set against SD$_0$ to SD$_5$ and according to different normalized score thresholds, with the basic and <SEP> methods. Again, we can see that the majority of the candidate sets obtain a recall of 0, although it is increased when the <SEP> method is used, when the normalized score threshold is 0 (all candidates are considered) and when the candidates are compared with SD$_1$. Indeed, that is when the largest number of candidate sets obtain a recall of 1 and the recall scores are most distributed.

3.3.4. F-score

The F-score combines the precision and recall results to offer a more global evaluation. We tweaked the precision and the recall presented above. Indeed, most of the precision and recall scores were 0, which prevented us from calculating the F-score for most of the candidate sets. To rectify this, we chose to replace every precision and recall score of 0 with a value of 0.0001. This way, the results were affected very little and we were able to calculate the F-score for all of the sets. Table 7 shows the F-score's micro-average with different SDs.

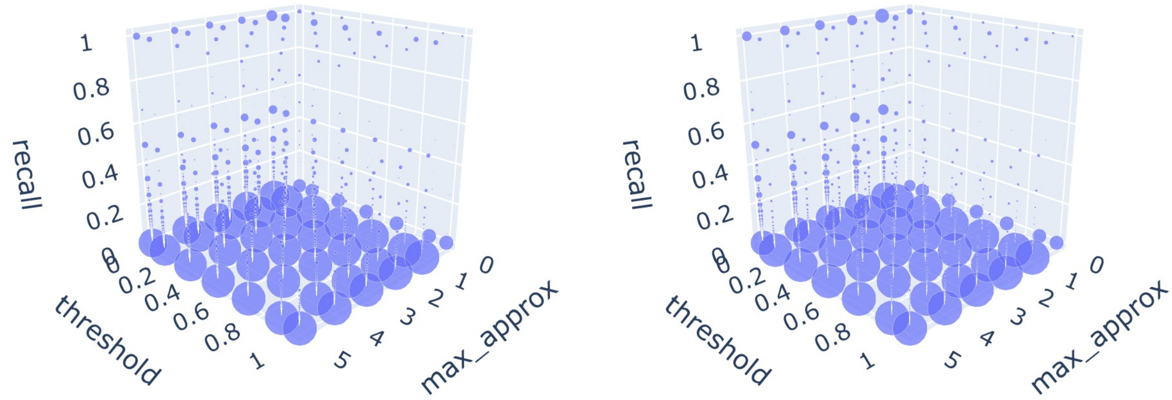
Figure 6 shows the F-scores we obtained against SD$_0$ to SD$_5$, with different normalized score thresholds and MAP values, for both methods. As expected, the F-score is also very low, and it is most distributed when the normalized score threshold is 0, especially when the <SEP> method was used.

3.3.5. Discussion

From this data, we conclude that CamemBERT is unable to reproduce the SD in a manner that would allow us to use the candidates to enrich it or to lexicalize new entries automatically. The <SEP> method slightly improves the results on all metrics. The certainty score does not seem to have a large impact on the quality of the candidates, since the higher results are obtained when all candidates are considered (threshold = 0) instead of only the one CamemBERT is most certain about (threshold = 1), although that could be due to the fact that only one possibility of an overlapping candidate divided by a large number of lexicalizations/candidates would still return a low recall/precision score, as we discuss below.

3.3.6. Candidates absent from the SD

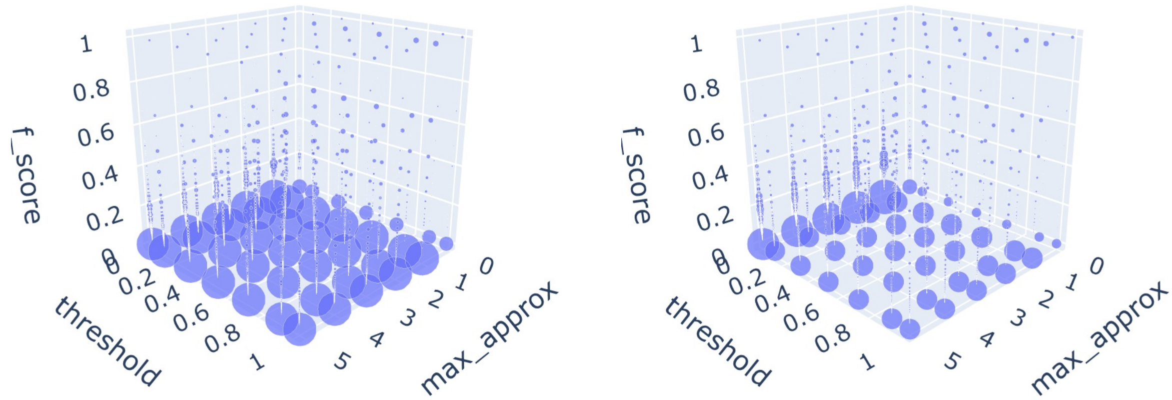
Since our primary objective was not to reproduce a SD we already have, but to broaden it, we also needed to evaluate the quality of the candidates suggested by CamemBERT that were not in the lexicalization set corresponding to the entry for which they had been generated. Thus, we evaluated manually a sample of those candidates. But before that, we wanted to verify that the normalized scores really did not have an impact on the candidates' quality. To do so, we looked at the proportion of candidates



(a) Basic method

(b) <SEP> method

Figure 5: Recall



(a) Basic method

(b) <SEP> method

Figure 6: F-score

Method	Score $\geq$ 0	Score $\geq$ 0.85
Basic	3.9 %	8.2 %
<SEP>	6.9 %	15.0 %

Table 8: % of candidates found in SD_1 according to normalized score and method

found in the SDs[5] (the precision) when their normalized score is not restricted, and then with a threshold of 0.85, which corresponds most of the time to the three candidates CamemBERT is the most certain of.

Table 8 shows the percentages obtained both with the basic method and the <SEP> method. As we can see, this test confirms that the <SEP> method is slightly superior to the basic method, since all results, no matter the normalized score threshold, are higher. It also shows us that the certainty score does have an impact, since with both methods, a larger proportion of the candidates with a higher score can be found in the SD than when all candidates are considered.

As such, since we had to evaluate a sample of thousands of generated candidates, we limited ourselves to the candidates with a normalized score $\geq$ 0.85, which, according to our evaluation, are of slightly better quality than those with a lower score. We evaluated whether an absent candidate should be included in the lexicalization set associated to the entry it had been generated for in a binary fashion. We evaluated them against SD_0 and SD_1 only, because we found that those categories were linguistically intuitive, i.e., the exact same meaning and approximately the same meaning, respectively. Deciding which lexicalizations should be part of SDs with higher approximation degrees felt arbitrary and too abstract for us to have any intuition, so we refrained from evaluating whether the candidates should be included in them. For each SD, we sampled 500 candidates, which corresponds

[5]We tested with SD_0 to SD_5 but only report the results for SD_1; they were very similar for every SD.

Entry	Lexicalization	Candidate	Syn
bouffer[1] IV.3 'chow down'	*manger_v* VII 'eat'	*parler* 'speak'	F
dégueulasse VI.3 'disgusting'	*dégueu* VI.1 'gross'	*ignoble* 'vile'	T
s'arrêter III.2 'stop oneself'	*arrêter* VI 'stop'	*cesser* 'cease'	T

Table 9: Extract of evaluation table for candidates absent from SD_0.

Normalized score	SD 0	SD 1
$\geq$ 0.85 but $<$ 1	2.0 %	12.0 %
= 1	5.6 %	10.0 %
Total	7.6 %	22.0 %

Table 10: % of absent candidates that should be included in the SDs (basic method)

Normalized score	SD 0	SD 1
$\geq$ 0.85 but $<$ 1	6.5 %	14.8 %
= 1	9.6 %	22.0 %
Total	16.1 %	36.8 %

Table 11: % of absent candidates that should be included in the SDs (<SEP> method)

to about 1 % of the number of sentences CamemBERT generated candidates for (around 45,000). For each SD, half of those 500 candidates had a normalized score below 1 (but at least 0.85), and half had a normalized score of exactly 1. These categories correspond roughly to the second, third or fourth most likely candidate, and the first one, respectively.

Table 9 shows a few examples of the candidates we kept and whether we felt they should be included in the associated lexicalization set. Tables 10 and 11 show the percentage of candidates we considered should be included in the corresponding SD based on the method used to generate them. We found that, although the proportion of absent candidates that should be included in the SD is more encouraging than CamemBERT's ability to reproduce it, it is still not high enough to consider that every candidate with a high normalized score is good enough to be systematically included in the SD. We also confirmed again that the <SEP> method was superior to the basic method, with a higher proportion of candidates that should be included, no matter the score or the SD they should be in. Yet, more than a third of the sampled candidates are good enough to be included in the SD 1, which is non-negligible. Thus, a tool harnessing CamemBERT's suggestions for new entries as lexicographers expand the SD could be useful.

4. Conclusion

Only one of the two methods presented in §1 succeeded. We managed to successfully build a SD compatible with GenDR automatically with the data contained in the LN-fr, and we used a parameter, MAP, to specify the degree of semantic approximation of the SD. This method hinges on the identification of a subset of lexical function (LF) called semantically empty paradigmatic lexical function (SEPLF), which to our knowledge had never been discussed in the literature. Thus, our SD contains 29,399 entries and as many unique LUs. When $MAP = 0$, these LUs are linked through 49,235 lexicalization links, and when $MAP = 5$, they are linked through 572,686 lexicalization and quasi-lexicalization links, considerably broadening GenDR's original French SD. This enables the realizer to produce many paraphrases from a given input.

Then, we generated the SD entries' closest neighbours in context to broaden it with CamemBERT using two methods: the so-called basic and <SEP> methods. We found that the <SEP> method was slightly superior in terms of the precision, recall and F-score obtained by the candidate sets calculated against the corresponding lexicalization sets found in the SDs produced from LN-fr. Through these metrics, we found that CamemBERT is unable to reproduce the SD_0 to SD_5, which leads us to believe that very few candidates have a good enough quality to improve the SD. Nonetheless, we evaluated whether a sample of the candidates absent from the SD_0 and SD_1 with the highest normalized scores should be included in the lexicalizations set associated to the entry for which said candidates had been generated. We found that with the <SEP> method, around 16 % of the sample should be included in SD_0, and around 37 % should be included in the SD_1, which is non-negligible. Thus, we think that a tool harnessing CamemBERT's suggestions to support lexicographers would be useful, but we conclude that the candidates cannot be added systematically and automatically to the SD without introducing too much noise.

Finally, we believe it would be interesting to reproduce this research with more recent language models. Indeed, when we started this study, CamemBERT had just come out and was state-of-the-art for French, whereas today, many similar models obtain better results and have been trained on more data. This leads us to believe that the possibility to expand a SD automatically through language models could be achieved someday.

5. Appendix

Name	Type	Approx	Same POS
Syn	paradigmatic	0	T
Syn_⊂	paradigmatic	1	T
Syn_⊃	paradigmatic	1	T
Syn_∩	paradigmatic	1	T
Syn_⊃^sex	paradigmatic	1	T
Syn_⊂^sex	paradigmatic	1	T
Conv_2	paradigmatic	0	T
Conv_21	paradigmatic	0	T
Conv_213	paradigmatic	0	T
Conv_23	paradigmatic	0	T
Conv_231	paradigmatic	0	T
Conv_312	paradigmatic	0	T
Conv_32	paradigmatic	0	T
Conv_321	paradigmatic	0	T
Conv_3214	paradigmatic	0	T
Conv_423	paradigmatic	0	T
Gener	paradigmatic	1	F
Figur	paradigmatic	0	F
S_0	paradigmatic	0	F
S_0_⊂	paradigmatic	1	F
S_0_⊃	paradigmatic	1	F
S_0_∩	paradigmatic	1	F
S_1 ∨ S_0	paradigmatic	0	F
S_2 ∨ S_0	paradigmatic	0	F
S_3 ∨ S_0	paradigmatic	0	F
S_4 ∨ S_0	paradigmatic	0	F
S_0Conv_21	paradigmatic	0	F
S_0Pred	paradigmatic	0	F
S_0Pred_⊂	paradigmatic	1	F
S_0Pred_∩	paradigmatic	1	F
S_0^usual	paradigmatic	1	F
S_0Conv_213	paradigmatic	0	F
V_0	paradigmatic	0	F
V_0_⊃	paradigmatic	1	F
V_0_⊂	paradigmatic	1	F
V_0_∩	paradigmatic	1	F
V_0Conv_21	paradigmatic	0	F
V_0Conv_213	paradigmatic	0	F
V_0Conv_413	paradigmatic	0	F
A_0	paradigmatic	0	F
A_0_⊃	paradigmatic	1	F
A_0_⊂	paradigmatic	1	F
A_0_∩	paradigmatic	1	F
Adv_0 ∨ A_0	paradigmatic	0	F
Adv_0	paradigmatic	0	F
Adv_0_∩	paradigmatic	1	F
Adv_0_⊃	paradigmatic	1	F
A_1	paradigmatic	0	F
A_1_⊂	paradigmatic	1	F
A_1_⊃	paradigmatic	1	F
A_1_∩	paradigmatic	1	F
A_1/2_⊂	paradigmatic	1	F
Adv_1 ∨ A_1	paradigmatic	0	F
A_1Pred	paradigmatic	0	F
S_0A_1	paradigmatic	0	F
A_2	paradigmatic	0	F
A_2_∩	paradigmatic	1	F
Adv_2 ∨ A_2	paradigmatic	0	F
A_3	paradigmatic	0	F
A_4	paradigmatic	0	F
Adv_1	syntagmatic	0	F
Adv_2	syntagmatic	0	F
Pred	syntagmatic	0	F
Conv_21Pred	syntagmatic	0	F

Table 12: All SEPLFs extracted from LN-fr

6. Bibliographical References

Juri Apresjan. 2000. *Systematic lexicography*. Oxford linguistics. Oxford University Press, Oxford; New York.

Bernd Bohnet, Andreas Langjahr, and Leo Wanner. 2000. A development environment for an MTT-based sentence generator. In *Proceedings of the first international conference on Natural language generation - INLG '00*, volume 14, page 260, Mitzpe Ramon, Israel. Association for Computational Linguistics.

Bernd Bohnet and Leo Wanner. 2010. Open soucre graph transducer interpreter and grammar development environment. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC) 2010*, pages 17–23, Valletta, Malte. European Language Resources Association.

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.

Ieva Dubinskaitė. 2017. Développement de ressources lituaniennes pour un générateur automatique de texte multilingue. Master's thesis, Université de Montréal.

J. R. Firth. 1957. A Synopsys of Linguistic Theory 1930-1955. In *Studies in Linguistic Analysis*, volume 1. Philological Society, Oxford. Réimprimé dans Palmer, F. (éd. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

Daniel Galarreta-Piquette. 2018. Intégration de VerbNet dans un réalisateur profond. Master's thesis, Université de Montréal.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(2–3):146–162.

Sylvain Kahane. 2003. The Meaning-Text Theory. *Dependency and Valency. An International Handbook of Contemporary Research*, 1:546–570.

François Lareau, Florie Lambrey, Ieva Dubinskaitė, Daniel Galarreta-Piquette, and Maryam Nejat. 2018. GenDR: A generic deep realizer with complex lexicalization. In *Proceedings of 11th edition of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japon.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Eric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Igor Mel'čuk. 1995. The future of the lexicon in linguistic description and the explanatory combinatorial dictionary. *Linguistics in the morning calm*, 3:181–270.

Igor Mel'čuk. 1996. Lexical functions: A tool for the description of lexical relations in the lexicon. In Leo Wanner, editor, *Lexical Functions in Lexicography and Natural Language Processing*. John Benjamins, Amsterdam.

Igor Mel'čuk. 2012. *Semantics: from Meaning to Text*. John Benjamins, Amsterdam/Philadelphia.

Igor Mel'čuk. 2016. *Language: from meaning to text*, Beck, David edition. LRC Publishing House/ Academic Studies Press, Moscow : Boston. OCLC: ocn932263842.

Igor Mel'čuk and Alain Polguère. 2021. Les fonctions lexicales dernier cri. In Sébastien Marengo, editor, *La Théorie Sens-Texte. Concepts-clés et applications*, Dixit Grammatica, pages 75–155. L'Harmattan.

Igor Mel'čuk. 1988. *Dependency syntax: theory and practice*. SUNY press, New York.

Igor Mel'čuk, Alain Polguère, and André Clas. 1995. *Introduction à la lexicologie explicative et combinatoire*. Duculot, Louvain-la-Neuve.

Jasmina Milićević. 2006. A Short Guide to the Meaning-Text Theory. *Journal of Koralex*, 8:187–233.

Sandrine Ollinger and Alain Polguère. 2020. Distribution des systèmes lexicaux. version 2.0. Technical report, ATILF—CNRS, Nancy, France.

Alain Polguère. 2009. Lexical systems: graph models of natural language lexicons. *Language Resources and Evaluation*, 43(1):41–55.

Alain Polguère. 2014. From writing dictionaries to weaving lexical networks. *International Journal of Lexicography*, 27(4):396–418.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2019. Lexical simplification with pretrained encoders. In *AAAI Conference on Artificial Intelligence*.

Benoît Sagot. 2010. The Lefff, a Freely Available and Large-coverage Morphological and Syntactic Lexicon for French. In *International Conference on Language Resources and Evaluation*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

Xiaoyu Zhao. 2018. Les collocations du champ sémantique des émotions en mandarin. Master's thesis, Université de Montréal, November.

Alexander Žolkovskij and Igor Mel'čuk. 1967. O sisteme semantičeskogo sinteza. II: Pravila preobrazovanij. *Naučno-texničeskaja informacija*, 2:17–27.