Let’s Play Mono-Poly: BERT Can Reveal Words’ Polysemy Level and Partitionability into Senses

Aina Gari Soler
Université Paris-Saclay
CNRS, LISN
91400, Orsay, France
aina.gari@limsi.fr

Marianna Apidianaki
Department of Digital Humanities
University of Helsinki
Helsinki, Finland
marianna.apidianaki@helsinki.fi

Abstract

Pre-trained language models (LMs) encode rich information about linguistic structure but their knowledge about lexical polysemy remains unclear. We propose a novel experimental setup for analyzing this knowledge in LMs specifically trained for different languages (English, French, Spanish, and Greek) and in multilingual BERT. We perform our analysis on datasets carefully designed to reflect different sense distributions, and control for parameters that are highly correlated with polysemy such as frequency and grammatical category. We demonstrate that BERT-derived representations reflect words’ polysemy level and their partitionability into senses. Polysemy-related information is more clearly present in English BERT embeddings, but models in other languages also manage to establish relevant distinctions between words at different polysemy levels. Our results contribute to a better understanding of the knowledge encoded in contextualized representations and open up new avenues for multilingual lexical semantics research.

1 Introduction

Pre-trained contextual language models have advanced the state of the art in numerous natural language understanding tasks (Devlin et al., 2019; Peters et al., 2018). Their success has motivated a large number of studies exploring what these models actually learn about language (Voita et al., 2019a; Clark et al., 2019; Voita et al., 2019b; Tenney et al., 2019). The bulk of this interpretation work relies on probing tasks that serve to predict linguistic properties from the representations generated by the models (Linzen, 2018; Rogers et al., 2020). The focus was initially put on linguistic aspects pertaining to grammar and syntax (Linzen et al., 2016; Hewitt and Manning, 2019; Hewitt and Liang, 2019). The first probing tasks addressing semantic knowledge explored phenomena in the syntax-semantics interface, such as semantic role labeling and coreference (Tenney et al., 2019; Kovaleva et al., 2019), and the symbolic reasoning potential of LM representations (Talmor et al., 2020).

Lexical meaning was largely overlooked in early interpretation work, but is now attracting increasing attention. Pre-trained LMs have been shown to successfully leverage sense annotated data for disambiguation (Wiedemann et al., 2019; Reif et al., 2019). The interplay between word type and token-level information in the hidden representations of LSTM LMs has also been explored (Aina et al., 2019), as well as the similarity estimates that can be drawn from contextualized representations without directly addressing word meaning (Ethayarajh, 2019). In recent work, Vulić et al. (2020) probe BERT representations for lexical semantics, addressing out-of-context word similarity. Whether these models encode knowledge about lexical polysemy and sense distinctions is, however, still an open question. Our work aims to fill this gap.

We propose methodology for exploring the knowledge about word senses in contextualized representations. Our approach follows a rigorous experimental protocol proper to lexical semantic analysis, which involves the use of datasets carefully designed to reflect different sense distributions. This allows us to investigate the knowledge models acquire during training, and the influence of context variation on token representations. We account for the strong correlation between word frequency and number of senses (Zipf, 1945), and for the relationship between grammatical category and polysemy, by balancing the frequency and part of speech...
Figure 1: BERT distinguishes monosemous (mono) from polysemous (poly) words in all layers. Representations for a poly word are obtained from sentences reflecting up to ten different senses (poly-bal), the same sense (poly-same), or natural occurrence in a corpus (poly-rand).

(PoS) distributions in our datasets and applying a frequency-based model to polysemy prediction.

Importantly, our investigation encompasses monolingual models in different languages (English, French, Spanish, and Greek) and multilingual BERT (mBERT). We demonstrate that BERT contextualized representations encode an impressive amount of knowledge about polysemy, and are able to distinguish monosemous (mono) from polysemous (poly) words in a variety of settings and configurations (cf. Figure 1). Importantly, we demonstrate that representations derived from contextual LMs encode knowledge about words’ polysemy acquired through pre-training which is combined with information from new contexts of use (Sections 3–6). Additionally, we show that BERT representations can serve to determine how easy it is to partition a word’s semantic space into senses (Section 7).

Our methodology can serve for the analysis of words and datasets from different topics, domains and languages. Knowledge about words’ polysemy and sense partitionability has numerous practical implications: It can guide decisions towards a sense clustering or a per-instance approach in applications (Reisinger and Mooney, 2010; Neelakantan et al., 2014; Camacho-Collados and Pilehvar, 2018); point to words with stable semantics which can be safe cues for disambiguation in running text (Leacock et al., 1998; Agirre and Martinez, 2004; Loureiro and Camacho-Collados, 2020); determine the needs in terms of context size for disambiguation (e.g., in queries, chatbots); help lexicographers define the number of entries for a word to be present in a resource, and plan the time and effort needed in semantic annotation tasks (McCarthy et al., 2016). It could also guide cross-lingual transfer, serving to identify polysemous words for which transfer might be harder. Finally, analyzing words’ semantic space can be highly useful for the study of lexical semantic change (Rosenfeld and Erk, 2018; Dubossarsky et al., 2019; Giulianelli et al., 2020; Schlechtweg et al., 2020). We make our code and datasets available to enable comparison across studies and to promote further research in these directions.

2 Related Work

The knowledge pre-trained contextual LMs encode about lexical semantics has only recently started being explored. Works by Reif et al. (2019) and Wiedemann et al. (2019) propose experiments using representations built from Wikipedia and the SemCor corpus (Miller et al., 1993), and show that BERT can organize word usages in the semantic space in a way that reflects the meaning distinctions present in the data. It is also shown that BERT can perform well in the word sense disambiguation (WSD) task by leveraging the sense-related information available in these resources. These works address the disambiguation capabilities of the model but do not show what BERT actually knows about words’ polysemy, which is the main axis of our work. In our experiments, sense annotations are not used to guide the models into establishing sense distinctions, but rather for creating controlled conditions that allow us to analyze BERT’s inherent knowledge of lexical polysemy.

Probing has also been proposed for lexical semantics analysis, but addressing different questions than the ones posed in our work. Aina et al., (2019) probe the hidden representations of a bidirectional (bi-LSTM) LM for lexical (type-level) and contextual (token-level) information. They specifically train diagnostic classifiers on the tasks of retrieving the input embedding of a word and a representation of its contextual

Our code and data are available at https://github.com/ainagari/monopoly.
meaning (as reflected in its lexical substitutes). The results show that the information about the input word that is present in LSTM representations is not lost after contextualization; however, the quality of the information available for a word is assessed through the model's ability to identify the corresponding embedding, as in Adi et al. (2017) and Conneau et al. (2018). Also, lexical ambiguity is only viewed through the lens of contextualization. In our work, on the contrary, it is given a central role: We explicitly address the knowledge BERT encodes about words' degree of polysemy and partitionability into senses. Vulić et al. (2020) also propose to probe contextualized models for lexical semantics, but they do so using "static" word embeddings obtained through pooling over several contexts, or extracting representations for words in isolation and from BERT's embedding layer, before contextualization. These representations are evaluated on tasks traditionally used for assessing the quality of static embeddings, such as out-of-context similarity and word analogy, which are not tailored for addressing lexical polysemy. Other contemporaneous work explores lexical polysemy in static embeddings (Jakubowski et al., 2020), and the relation of ambiguity and context uncertainty as approximated in the space constructed by mBERT using information-theoretic measures (Pimentel et al., 2020). Finally, work by Ethayarajh (2019) provides useful observations regarding the impact of context on the representations, without explicitly addressing the semantic knowledge encoded by the models. Through an exploration of BERT, ELMo, and GPT-2 (Radford et al., 2019), the author highlights the highly distorted similarity of the obtained contextualized representations which is due to the anisotropy of the vector space built by each model.2 The question of meaning is not addressed in this work, making it hard to draw any conclusions about lexical polysemy.

Our proposed experimental setup is aimed at investigating the polysemy information encoded in the representations built at different layers of deep pre-trained LMs. Our approach basically relies on the similarity of contextualized representations, which amounts to word usage similarity (Usim) estimation, a classical task in lexical semantics (Erk et al., 2009; Huang et al., 2012; Erk et al., 2013). The Usim task precisely involves predicting the similarity of word instances in context without use of sense annotations. BERT has been shown to be particularly good at this task (Garí Soler et al., 2019; Pilehvar and Camacho-Collados, 2019). Our experiments allow us to explore and understand what this ability is due to.

3 Lexical Polysemy Detection

3.1 Dataset Creation

We build our English dataset using SemCor 3.0 (Miller et al., 1993), a corpus manually annotated with WordNet senses (Fellbaum, 1998). It is important to note that we do not use the annotations for training or evaluating any of the models. These only serve to control the composition of the sentence pools that are used for generating contextualized representations, and to analyze the results. We form sentence pools for monosemous (mono) and polysemous (poly) words that occur at least ten times in SemCor.3 For each mono word, we randomly sample ten of its instances in the corpus. For each poly word, we form three sentence pools of size ten reflecting different sense distributions:

- **Balanced** (poly-bal). We sample a sentence for each sense of the word in SemCor until a pool of ten sentences is formed.
- **Random** (poly-rand). We randomly sample ten poly word instances from SemCor. We expect this pool to be highly biased towards a specific sense due to the skewed frequency distribution of word senses (Kilgarriff, 2004; McCarthy et al., 2004). This configuration is closer to the expected natural occurrence of senses in a corpus, it thus serves to estimate the behaviour of the models in a real-world setting.
- **Same sense** (poly-same). We sample ten sentences illustrating only one sense of the poly word. Although the composition of this pool is similar to that of the mono pool (i.e. all instances describe the same sense) we call it

---

2This issue affects all tested models and is particularly present in the last layers of GPT-2, resulting in highly similar representations even for random words.

3We find the number of senses for a word of a specific part of speech (PoS) in WordNet 3.0, which we access through the NLTK interface (Bird et al., 2009).
Table 1: Example sentences for the monosemous noun *hotel* and the polysemous noun *room*.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Word</th>
<th>Sense</th>
<th>Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>mono</td>
<td>hotel.n</td>
<td>INN</td>
<td>The walk ended, inevitably, right in front of his hotel building. Maybe he’s at the hotel.</td>
</tr>
<tr>
<td>poly-same</td>
<td>room.n</td>
<td>CHAMBER</td>
<td>The room vibrated as if a giant hand had rocked it.</td>
</tr>
<tr>
<td>poly-bal</td>
<td>room.n</td>
<td>SPACE</td>
<td>(. . .) he left the room, walked down the hall (. . .)</td>
</tr>
<tr>
<td></td>
<td>room.n</td>
<td>OPPORTUNITY</td>
<td>It gives them room to play and plenty of fresh air.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(. . .) Tell her to come to Adam’s room (. . .)</td>
</tr>
</tbody>
</table>

poly-same because it describes one sense of a polysemous word. Specifically, we want to explore whether BERT representations derived from these instances can serve to distinguish mono from poly words.

The controlled composition of the poly sentence pools allows us to investigate the behavior of the models when they are exposed to instances of polysemous words describing the same or different senses. There are 1,765 poly words in SemCor with at least 10 sentences available. We randomly subsample 418 from these in order to balance the mono and poly classes. Our English dataset is composed of 836 mono and poly words, and their instances in 8,195 unique sentences. Table 1 shows a sample of the sentences in different pools. For French, Spanish, and Greek, we retrieve sentences from the Eurosense corpus (Delli Bovi et al., 2017) which contains texts from Europarl automatically annotated with BabelNet word senses (Navigli and Ponzetto, 2012). We extract sentences from the high-precision version of Eurosense, and create sentence pools in the same way as in English, balancing the number of monosemous and polysemous words (418). We determine the number of senses for a word as the number of its BabelNet senses that are mapped to a WordNet sense.

### 3.2 Contextualized Word Representations

We experiment with representations generated by three English models: BERT (Devlin et al., 2019), ELMo (Peters et al., 2018), and context2vec (Melamud et al., 2016). BERT is a Transformer architecture (Vaswani et al., 2017) that is jointly trained for a masked LM and a next sentence prediction task. Masked LM involves a Cloze-style task, where the model needs to guess randomly masked words by jointly conditioning on their left and right context. We use the bert-base-uncased and bert-base-cased models, pre-trained on the BooksCorpus (Zhu et al., 2015) and English Wikipedia. ELMo is a bi-LSTM LM trained on Wikipedia and news crawl data from WMT 2008-2012. We use 1024-d representations from the 5.5B model. Context2vec is a neural model based on word2vec’s CBOW architecture (Mikolov et al., 2013) which learns embeddings of wide sentential contexts using a bi-LSTM. The model produces representations for words and their context. We use the context representations from a 600-d context2vec model pre-trained on the ukWaC corpus (Baroni et al., 2009).

For French, Spanish, and Greek, we use BERT models specifically trained for each language: Flaubert (flaubert_base_uncased) (Le et al., 2020), BETO (bert-base-spanish-wwm-uncased) (Cañete et al., 2020), and Greek BERT (bert-base-greek-uncased-v1) (Koutsikakis et al., 2020). We also use the bert-base-multilingual-cased model for each of the four languages. mBERT was trained on...
Wikipedia data of 104 languages. All BERT models generate 768-d representations.

3.3 The Self-Similarity Metric

All models produce representations that describe word meaning in specific contexts of use. For each instance \( i \) of a target word \( w \) in a sentence, we extract its representation from: (i) each of the 12 layers of a BERT model; \(^{13}\) (ii) each of the three ELMo layers; and (iii) context2vec. We calculate self-similarity \((SelfSim)\) (Ethayarajh, 2019) for \( w \) in a sentence pool \( p \) and a layer \( l \), by taking the average of the pairwise cosine similarities of the representations of its instances in \( l \):

\[
SelfSim_l(w) = \frac{1}{|I|^2 - |I|} \sum_{i \in I} \sum_{j \in I, j \neq i} \cos(x_{wli}, x_{wlj})
\]

In formula 1, \(|I|\) is the number of instances for \( w \) (ten in our experiments); \( x_{wli} \) and \( x_{wlj} \) are the representations for instances \( i \) and \( j \) of \( w \) in layer \( l \). The \( SelfSim \) score is in the range \([-1, 1]\). We report the average \( SelfSim \) for all \( w \)'s in a pool \( p \).

We expect it to be higher for monosemous words and words with low polysemy than for highly polysemous words. We also expect the poly-same pool to have a higher average \( SelfSim \) score than the other poly pools which contain instances of different senses.

Contextualization has a strong impact on \( SelfSim \) since it introduces variation in the token-level representations, making them more dissimilar. The \( SelfSim \) value for a word would be 1 with non-contextualized (or static) embeddings, as all its instances would be assigned the same vector. In contextual models, \( SelfSim \) is lower in layers where the impact of the context is stronger (Ethayarajh, 2019). It is, however, important to note that contextualization in BERT models is not monotonic, as shown by previous studies of the models’ internal workings (Voita et al., 2019a; Ethayarajh, 2019). Our experiments presented in the next section provide additional evidence in this respect.

3.4 Results and Discussion

3.4.1 Mono-Poly in English

Figure 2 shows the average \( SelfSim \) value obtained for each sentence pool with representations produced by BERT models. The thin lines in the first plot illustrate the average \( SelfSim \) score calculated for mono and poly words using representations from each layer of the uncased English BERT model. We observe a clear distinction of words according to their polysemy: \( SelfSim \) is higher for mono than for poly words across all layers and sentence pools. BERT establishes a clear distinction even between the mono and poly-same pools, which contain instances of only one sense. This distinction is important; it suggests that BERT encodes information about a word’s monosemous or polysemous nature regardless of the sentences that are used to derive the contextualized representations. Specifically, BERT produces less similar representations for word instances in the poly-same pool compared to mono, reflecting that poly words can have different meanings.

We also observe a clear ordering of the three poly sentence pools: Average \( SelfSim \) is higher in poly-same, which only contains instances of one sense, followed by mid-range values in poly-rand, and gets its lowest values in the balanced setting (poly-bal). This is noteworthy given that poly-rand contains a mix of senses but with a stronger representation of \( w \)’s frequent sense than poly-bal (71% vs. 47%).\(^{14}\)

Our results demonstrate that BERT representations encode two types of lexical semantic knowledge: information about the polysemous nature of words acquired through pre-training (as reflected in the distinction between mono and poly-same), and information from the particular instances of a word used to create the contextualized representations (as shown by the finer-grained distinctions between different poly settings). BERT’s knowledge about polysemy can be due to differences in the types of context where words of different polysemy levels occur. We expect poly words to be seen in more varied contexts than mono words, reflecting their different senses. BERT encodes this variation with the

\(^{12}\)The mBERT model developers recommend using the cased version of the model rather than the uncased one, especially for languages with non-Latin alphabets, because it fixes normalization issues. More details about this model can be found here: https://github.com/google-research/bert/blob/master/multilingual.md.

\(^{13}\)We also tried different combinations of the last four layers, but this did not improve the results. When a word is split into multiple wordpieces (WPs), we obtain its representation by averaging the WPs.

\(^{14}\)Numbers are macro-averages for words in the pools.
LM objective through exposure to large amounts of data, and this is reflected in the representations. The same ordering pattern is observed with mBERT (lower part of Figure 2) and with ELMo (Figure 3(a)). With context2vec, average $SelfSim$ in mono is 0.40, 0.38 in poly–same, 0.37 in poly–rand, and 0.35 in poly–bal. This suggests that these models also have some inherent knowledge about lexical polysemy, but differences are less clearly marked than in BERT.

Using the cased model leads to an overall increase in $SelfSim$ and to smaller differences between bands, as shown by the thick lines in the first plot of Figure 2. Our explanation for the lower distinction ability of the bert–base–cased model is that it encodes sparser information about words than the uncased model. It was trained on a more diverse set of strings, so many WPs are present in both their capitalized and non-capitalized form in the vocabulary. In spite of that, it has a smaller vocabulary size (29K WPs) than the uncased model (30.5K). Also, a higher number of WPs correspond to word parts than in the uncased model (6,478 vs 5,829).

We test the statistical significance of the mono/poly–rand distinction using unpaired two-samples $t$-tests when the normality assumption is met (as determined with Shapiro Wilk’s tests). Otherwise, we run a Mann Whitney U test, the non-parametrical alternative of this $t$-test. In order to lower the probability of type I errors (false positives) that increases when performing multiple tests, we correct $p$-values using the Benjamini–Hochberg False Discovery Rate (FDR) adjustment (Benjamini and Hochberg, 1995). Our results show that differences are significant across all embedding types and layers ($\alpha = 0.01$).

The decreasing trend in $SelfSim$ observed for BERT in Figure 2, and the peak in layer 11, confirm the phases of context encoding and token reconstruction observed by Voita et al. (2019a).15 In earlier layers, context variation makes representations more dissimilar and $SelfSim$ decreases. In the last layers, information about the input token is recovered for LM prediction and similarity scores are boosted. Our results show clear distinctions across all BERT and ELMo layers. This suggests that lexical information is spread throughout the layers of the models, and contributes new evidence to the discussion on the localization of semantic information inside the models (Rogers et al., 2020; Vulić et al., 2020).

![Figure 2: Average $SelfSim$ obtained with monolingual BERT models (top row) and mBERT (bottom row) across all layers (horizontal axis). In the first plot, thick lines correspond to the cased model.](image)

![Figure 3: Comparison of average $SelfSim$ obtained for mono and poly words using ELMo representations (a), and for words in different polysemy bands in the poly–rand sentence pool (b).](image)

15They study the information flow in the Transformer estimating the MI between representations at different layers.
3.4.2 Mono–Poly in Other Languages

The top row of Figure 2 shows the average SelfSim obtained for French, Spanish, and Greek words using monolingual models. Flaubert, BETO, and Greek BERT representations clearly distinguish mono and poly words, but average SelfSim values for different poly pools are much closer than in English. BETO seems to capture these fine-grained distinctions slightly better than the French and Greek models. The second row of the figure shows results obtained with mBERT representations. We observe the highly similar average SelfSim values assigned to different poly pools, which show that distinction is harder than in monolingual models. Statistical tests show that the difference between SelfSim values in mono and poly-rand is significant in all layers of BETO, Flaubert, Greek BERT, and mBERT for Spanish and French.16

The magnitude of the difference in Greek BERT is, however, smaller compared to the other models (0.03 vs. 0.09 in BETO at the layers with the biggest difference in average SelfSim).

4 Polysemy Level Prediction

4.1 SelfSim-based Ranking

In this set of experiments, we explore the impact of words’ degree of polysemy on the representations. We control for this factor by grouping words into three polysemy bands as in McCarthy et al. (2016), which correspond to a specific number of senses (k): low: \(2 \leq k \leq 3\), mid: \(4 \leq k \leq 6\), high: \(k > 6\). For English, the three bands are populated with a different number of words: low: 551, mid: 663, high: 551. In the other languages, we form bands containing 300 words each.17 In Figure 4, we compare mono words with words in each polysemy band in terms of their average SelfSim. Values for mono words are taken from Section 3. For poly words, we use representations from the poly-rand sentence pool, which better approximates natural occurrence in a corpus. For comparison, we report in Figure 5 results obtained in English using sentences from the poly-same and poly-bal pools.18

In English, the pattern is clear in all plots: SelfSim is higher for mono than for poly words in any band, confirming that BERT is

16In mBERT for Greek, the difference is significant in ten layers.

17We only used 418 of these poly words in Section 3 in order to have balanced mono and poly pools.

18We omit the plots for poly-bal and poly-same for the other models due to space constraints.
able to distinguish mono from poly words at different polysemy levels. The range of SelfSim values for a band is inversely proportional to its \( k \): Words in low get higher values than words in high. The results denote that the meaning of highly polysemous words is more variable (lower SelfSim) than the meaning of words with fewer senses. As expected, scores are higher and inter-band similarities are closer in poly-same (cf. Figure 5(b)) compared with poly-rand and poly-bal, where distinctions are clearer. The observed differences confirm that BERT can predict the polysemy level of words, even from instances describing the same sense.

We observe similar patterns with ELMo (cf. Figure 3(b)) and context2vec representations in poly-rand,\(^{19}\) but smaller absolute inter-band differences. In poly-same, both models fail to correctly order the bands. Overall, our results highlight that BERT encodes higher quality knowledge about polysemy. We test the significance of the inter-band differences detected in poly-rand using the same approach as in Section 3.4.1. These are significant in all but a few\(^{20}\) layers of the models.

The bands are also correctly ranked in the other three languages but with smaller inter-band differences than in English, especially in Greek where clear distinctions are only made in a few middle layers. This variation across languages can be explained to some extent by the quality of the automatic EuroSense annotations, which has a direct impact on the quality of the sentence pools. Results of a manual evaluation conducted by Delli Bovi et al. (2017) showed that WSD precision is ten points higher in English (81.5) and Spanish (82.5) than in French (71.8). The Greek portion, however, has not been evaluated.

Plots in the second row of Figure 4 show results obtained using mBERT. Similarly to the previous experiment (Section 3.4), mBERT overall makes less clear distinctions than the monolingual models. The low and mid bands often get similar SelfSim values, which are close to mono in French and Greek. Still, inter-band differences are significant in most layers of mBERT and the monolingual French, Spanish, and Greek models.\(^{21}\)

### 4.2 Anisotropy Analysis

In order to better understand the reasons behind the smaller inter-band differences observed with mBERT, we conduct an additional analysis of the models’ anisotropy. We create 2,183 random word pairs from the English mono, low, mid and high bands, and 1,318 pairs in each of the other languages.\(^{22}\) We calculate the cosine similarity between two random instances of the words in each pair and take the average over all pairs (RandSim). The plots in the left column of Figure 6 show the results. We observe a clear difference in the scores obtained by monolingual models (solid lines) and mBERT (dashed lines). Clearly, mBERT assigns higher similarities to

\[\text{\(19\)Average SelfSim values for context2vec in the poly-rand setting: low: 0.37, mid: 0.36, high: 0.36.}\]

\[\text{\(20\)low \rightarrow mid in ELMo’s third layer, and mid \rightarrow high in context2vec and in BERT’s first layer.}\]

\[\text{\(21\)With the exception of mono \rightarrow low in mBERT for Greek, and low \rightarrow mid in Flaubert and in mBERT for French.}\]

\[\text{\(22\)1,318 is the total number of words across bands in French, Spanish, and Greek.}\]
random words, an indication that its semantic space is more anisotropic than the one built by monolingual models. High anisotropy means that representations occupy a narrow cone in the vector space, which results in lower quality similarity estimates and in the model’s limited potential to establish clear semantic distinctions.

We also compare RandSim to the average SelfSim obtained for poly words in the poly-rand sentence pool (cf. Section 3.1). In a quality semantic space, we would expect SelfSim (between same word instances) to be much higher than RandSim. The right column of Figure 6 shows the difference between these two scores. diff in a layer \( l \) is calculated as in Equation (2):

\[
\text{diff}_l = \text{AvgSelfSim}_l(\text{poly-rand}) - \text{RandSim}_l
\]

We observe that the difference is smaller in the space built by mBERT, which is more anisotropic than the space built by monolingual models. This is particularly obvious in the upper layers of the model. This result confirms the lower quality of mBERT’s semantic space compared to monolingual models.

Finally, we believe that another factor behind the worse mBERT performance is that the multilingual WP vocabulary is mostly English-driven, resulting in arbitrary partitionings of words in the other languages. This word splitting procedure must have an impact on the quality of the lexical information in mBERT representations.

5 Analysis by Frequency and PoS

Given the strong correlation between word frequency and number of senses (Zipf, 1945), we explore the impact of frequency on BERT representations. Our goal is to determine the extent to which it influences the good mono/poly detection results obtained in Sections 3.4 and 4.1.

5.1 Dataset Composition

We perform this analysis in English using frequency information from Google Ngrams (Brants and Franz, 2006). For French, Spanish, and Greek, we use frequency counts gathered from the OSCAR corpus (Suárez et al., 2019). We split the words into four ranges \( F \) corresponding to the quartiles of frequencies in each dataset. Each range \( f \) in \( F \) contains the same number of words. We provide detailed information about the composition of the English dataset in Figure 7.\(^{23}\) Figure 7(a) shows that mono words are much less frequent than poly words. Figure 7(b) shows the distribution of different PoS categories in each band. Nouns are the prevalent category in all bands and verbs are less present among mono words (10.8%), as expected. Finally, adverbs are hardly represented in the high polysemy band (1.2% of all words).

5.2 Self-Sim by Frequency Range and PoS

We examine the average BERT SelfSim per frequency range in poly-rand. Due to space constraints, we only report detailed results for the English BERT model in Figure 8 (plot (a)). The clear ordering by range suggests that BERT can successfully distinguish words by their frequency, especially in the last layers. Plot (b) in Figure 8 shows the average SelfSim for words of each PoS category. Verbs have the lowest SelfSim which is not surprising given that they are highly polysemous (as shown in Figure 7(b)). We observe the same trend for monolingual models in the other three languages.

\(^{23}\)The composition of each band is the same as in Sections 3 and 4.
5.3 Controlling for Frequency and PoS

We conduct an additional experiment where we control for the composition of the poly bands in terms of grammatical category and word frequency. We call these two settings POS-bal and FREQ-bal. We define $n_{\text{pos}}$, the smallest number of words of a specific PoS that can be found in a band. We form the POS-bal bands by subsampling from each band the same number of words ($n_{\text{pos}}$) of that PoS. For example, all POS-bal bands have $n_{\text{nouns}}$ nouns and $n_{\text{verbs}}$ verbs. We follow a similar procedure to balance the bands by frequency in the FREQ-bal setting. In this case, $n_{f}$ is the minimum number of words of a specific frequency range $f$ that can be found in a band. We form the FREQ-bal dataset by subsampling from each band the same number of words ($n_{f}$) of a given range $f$ in $F$.

Table 2 shows the distribution of words per PoS and frequency range in the POS-bal and FREQ-bal settings. All bands for a language contain the same number of words of a specific grammatical category or frequency range. M stands for a million and m for a thousand occurrences of a word in a corpus.

![Figure 8: Average SelfSim for English words of different frequencies and part of speech categories with BERT representations.](image)

<table>
<thead>
<tr>
<th>Language</th>
<th>POS-bal</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>en</td>
<td>198</td>
<td>45</td>
<td>64</td>
<td>7</td>
</tr>
<tr>
<td>fr</td>
<td>171</td>
<td>32</td>
<td>29</td>
<td>9</td>
</tr>
<tr>
<td>es</td>
<td>167</td>
<td>25</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>FREQ-bal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>en</td>
<td>7.1M</td>
<td>20M</td>
<td>49M</td>
<td>682M</td>
</tr>
<tr>
<td>fr</td>
<td>23m</td>
<td>70m</td>
<td>210m</td>
<td>41M</td>
</tr>
<tr>
<td>es</td>
<td>64m</td>
<td>233m</td>
<td>793m</td>
<td>59M</td>
</tr>
<tr>
<td>el</td>
<td>14m</td>
<td>40m</td>
<td>111m</td>
<td>1.9M</td>
</tr>
</tbody>
</table>

Statistical tests show that all inter-band distinctions established by English BERT are still significant in most layers of the model. This is not the case for ELMo and context2vec, which can distinguish between mono and poly words but fail to establish significant distinctions between polysemy bands in the balanced settings. For French and Spanish, the statistical analysis shows that all distinctions in POS-bal are significant in at least one layer of the models. The same applies to the mono→poly distinction in FREQ-bal but finer-grained distinctions disappear.

6 Classification by Polysemy Level

Our finding that word instance similarity differs across polysemy bands suggests that this feature can be useful for classification. In this section, we probe the representations for polysemy using a classification experiment where we test their ability to guess whether a word is polysemous, and which poly band it falls in. We use

---

24Note that the sample size in this analysis is smaller compared to that used in Sections 3.4 and 4.1.

25With a few exceptions: mono→low and mid→high are significant in all BETO layers.
the poly-rand sentence pools and a standard train/dev/test split (70%/15%/15%) of the data.
For the mono/poly distinction (i.e., the data used in Section 3), this results in 584/126/126 words per set in each language. To guarantee a fair evaluation, we make sure there is no overlap between the lemmas in the three sets. We use two types of features: (i) the average SelfSim for a word; and (ii) all pairwise cosine similarities collected for its instances, which results in 45 features per word (pairCos). We train a binary logistic regression classifier for each type of representation and feature.

As explained in Section 4, the three poly bands (low, mid, and high) and mono contain a different number of words. For classification into polysemy bands, we balance each class by randomly subsampling words from each band. In total, we use 1,168 words for training, 252 for development, and 252 for testing (70%/15%/15%) in English. In the other languages, we use a split of 840/180/180 words. We train multi-class logistic regression classifiers with the two types of features, SelfSim and pairCos. We compare the results of the classifiers to a baseline that predicts always the same class and a classifier that only uses log frequency as feature. Subscripts denote the layers used.

Table 3 presents classification accuracy on the test set. We report results obtained with the best layer for each representation type and feature as determined on the development sets. In English, best accuracy is obtained by BERT in both the binary (0.79) and multiclass settings (0.49), followed by mBERT (0.77 and 0.46). Despite its simplicity, the frequency-based classifier obtains better results than context2vec and ELMo, and performs on par with mBERT in the binary setting. This shows that frequency information
is highly relevant for the mono-poly distinction. All classifiers outperform the same class baseline. These results are very encouraging, showing that BERT embeddings can be used to determine whether a word has multiple meanings, and provide a rough indication of its polysemy level. Results in the other three languages are not as high as those obtained in English, but most models give higher results than the frequency-based classifier.\footnote{Only exceptions are Greek mBERT in the multi-class setting, and Flaubert in both settings.}

7 Word Sense Clusterability

We have shown that representations from pre-trained LMs encode rich information about words’ degree of polysemy. They can successfully distinguish mono from poly lemmas, and predict the polysemy level of words. Our previous experiments involved a set of controlled settings representing different sense distributions and polysemy levels. In this section, we explore whether these representations can also point to the clusterability of poly words in an uncontrolled setting.

7.1 Task Definition

Instances of some poly words are easier to group into interpretable clusters than others. This is, for example, a simple task for the ambiguous noun rock which can express two clearly separate senses (stone and music), but harder for book, which might refer to the content or object senses of the word (e.g., I read a book vs. I bought a book). In what follows, we test the ability of contextualized representations to estimate how easy this task is for a specific word, that is, its partitionability into senses.

Following McCarthy et al. (2016), we use the clusterability metrics proposed by Ackerman and Ben-David (2009) to measure the ease of clustering word instances into senses. McCarthy et al. base their clustering on the similarity of manual meaning-preserving annotations (lexical substitutes and translations). Instances of different senses, such as: Put granola bars in a bowl vs. That’s not a very high bar, present no overlap in their in-context substitutes: \{snack, biscuit, block, slab\} vs. \{pole, marker, hurdle, barrier, level, obstruction\}. Semantically related instances, on the contrary, share a different number of substitutes depending on their proximity. The need for manual annotations, however, constrains the method’s applicability to specific datasets.

We propose to extend and scale up the McCarthy et al. (2016) clusterability approach using contextualized representations, in order to make it applicable to a larger vocabulary. These experiments are carried out in English due to the lack of evaluation data in other languages.

7.2 Data

We run our experiments on the usage similarity (Usim) dataset (Erk et al., 2013) for comparison with previous work. Usim contains ten instances for 56 target words of different PoS from the SemEval Lexical Substitution dataset (McCarthy and Navigli, 2007). Word instances are manually annotated with pairwise similarity scores on a scale from 1 (completely different) to 5 (same meaning).

We represent target word instances in Usim in two ways: using contextualized representations generated by BERT, context2vec, and ELMo (BERT-REP, c2v-REP, ELMo-REP);\footnote{We do not use the first layer of ELMo in this experiment.} using substitute-based representations with automatically generated substitutes. The substitute-based approach allows for a direct comparison with the method of McCarthy et al. (2016). They represent each instance \(i\) of a word \(w\) in Usim as a vector \(\vec{i}\). We generate our substitute-based representations in two ways: using contextualized representations with BERT using the simple “word similarity” approach in Zhou et al. (2019). For an instance \(i\) of word \(w\) in context \(C\), we rank a set of candidate substitutes \(S = \{s_1, s_2, \ldots, s_n\}\) based on the cosine similarity of the BERT representations for \(i\) and for each substitute \(s_j \in S\) in the same context \(C\). We use representations from the last layer of the model. As candidate substitutes, we use the unigram paraphrases of \(w\) in the Paraphrase
7.3 Sense Clustering

The clusterability metrics that we use are metrics initially proposed for estimating the quality of the optimal clustering that can be obtained from a dataset; the better the quality of this clustering, the higher the clusterability of the dataset it is derived from (Ackerman and Ben-David, 2009).

In order to estimate the clusterability of a word \( w \), we thus need to first cluster its instances in the data. We use the \( k \)-means algorithm which requires the number of senses for a lemma. This is, of course, different for every lemma in our dataset. We define the optimal number of clusters \( k \) for a lemma in a data-driven manner using the Silhouette coefficient \( (\text{sil}) \) (Rousseeuw, 1987), without recourse to external resources.\footnote{We use PPDB (http://www.paraphrase.org) to reduce variability in our substitute sets, compared to the ones that would be proposed by looking at the whole vocabulary.}

For a data point \( i \), \( \text{sil} \) compares the intra-cluster distance (i.e., the average distance from \( i \) to every other data point in the same cluster) with the average distance of \( i \) to all points in its nearest cluster. The \( \text{sil} \) value for a clustering is obtained by averaging \( \text{sil} \) for all data points, and it ranges from \(-1 \) to \(1 \). We cluster each type of representation for \( w \) using \( k \)-means with a range of \( k \) values \((2 \leq k \leq 10)\), and retain the \( k \) of the clustering with the highest mean \( \text{sil} \). Additionally, since BERT representations’ cosine similarity correlates well with usage similarity (Gari Soler et al., 2019), we experiment with Agglomerative Clustering with average linkage directly on the cosine distance matrix obtained with BERT representations (BERT-AGG). For comparison, we also use Agglomerative Clustering on the gold usage similarity scores from Usim, transformed into distances (Gold-AGG).

7.4 Clusterability Metrics

We use in our experiments the two best performing metrics from McCarthy et al. (2016): Variance Ratio \( (\text{vr}) \) (Zhang, 2001) and Separability \( (\text{sep}) \) (Ostrovsky et al., 2012). \( \text{vr} \) calculates the ratio of the within- and between-cluster variance for a given clustering solution. \( \text{sep} \) measures the difference in loss between two clusterings with \( k + 1 \) and \( k \) clusters and its range is \([0,1]\). We use \( k \)-means’ sum of squared distances of data points to their closest cluster center as the loss. Details about these two metrics are given in Appendix A.\footnote{Note that the \( \text{vr} \) and \( \text{sep} \) metrics are not compatible with Gold-AGG which relies on Usim similarity scores, because we need vectors for their calculation. For BERT-AGG, we calculate \( \text{vr} \) and \( \text{sep} \) using BERT embeddings.}

We also experiment with \( \text{sil} \) as a clusterability metric, as it can assess cluster validity. For \( \text{vr} \) and \( \text{sil} \), a higher value indicates higher clusterability. The inverse applies to \( \text{sep} \).

We calculate Spearman’s \( \rho \) correlation between the results of each clusterability metric and two gold standard measures derived from Usim: \( \text{Uiaa} \) and \( \text{Umid} \). \( \text{Uiaa} \) is the inter-annotator agreement for a lemma in terms of average pairwise Spearman’s correlation between annotators’ judgments. Higher \( \text{Uiaa} \) values indicate higher clusterability, meaning that sense partitions are clearer and easier to agree upon. \( \text{Umid} \) is the proportion of mid-range judgments (between 2 and 4) assigned by annotators to all instances of a target word. It indicates how often usages do not have identical (5) or completely different (1) meaning. Therefore, higher \( \text{Umid} \) values indicate lower clusterability.

7.5 Results and Discussion

The clusterability results are given in Table 4. Agglomerative Clustering on the gold Usim similarity scores (Gold-AGG) gives best results on the \( \text{Uiaa} \) evaluation in combination with the \( \text{sil} \) clusterability metric \((\rho = 0.80)\). This is unsurprising, since \( \text{Uiaa} \) and \( \text{Umid} \) are derived from the same Usim scores. From our automatically generated representations, the strongest correlation with \( \text{Uiaa} \) (0.69) is obtained...
Table 4: Spearman’s $\rho$ correlation between automatic metrics and gold standard clusterability estimates. Significant correlations (where the null hypothesis $\rho = 0$ is rejected with $\alpha < 0.05$) are marked with *. The arrows indicate the expected direction of correlation for each metric. Subscripts indicate the layer that achieved best performance. The two strongest correlations obtained with each gold standard measure are in boldface.

![Figure 10: Spearman’s $\rho$ correlations between the gold standard Uiaa and Umid scores, and clusterability estimates obtained using Agglomerative Clustering on a cosine distance matrix of BERT representations.](image)

We present a per layer analysis of the correlations obtained with the best performing BERT representations (BERT-AGG) and the SIL metric in Figure 10. We report the absolute values of the correlation coefficient for a more straightforward comparison. For Uiaa, the higher layers of the model make the best predictions: Correlations increase monotonically up to layer 10, and then they show a slight decrease. Umid prediction shows a more irregular pattern: It peaks at layers 3 and 8, and decreases again in the last layers.

8 Conclusion

We have shown that contextualized BERT representations encode rich information about lexical polysemy. Our experimental results suggest that this high quality knowledge about words, which allows BERT to detect polysemy in different configurations and across all layers, is acquired during pre-training. Our findings hold for the English BERT as well as for BERT models in other languages, as shown by our experiments on French, Spanish, and Greek, and to a lesser extent for multilingual BERT. Moreover, English BERT representations can be used to obtain a good estimation of a word’s partitionability into senses. These results open up new avenues for research in multilingual semantic analysis, and we can consider various theoretical and application-related extensions for this work.

The polysemy and sense-related knowledge revealed by the models can serve to develop novel methodologies for improved cross-lingual alignment of embedding spaces and cross-lingual transfer, pointing to more polysemous (or less clusterable) words for which transfer might be
harder. Predicting the polysemy level of words can also be useful for determining the context needed for acquiring representations that properly reflect the meaning of word instances in running text. From a more theoretical standpoint, we expect this work to be useful for studies on the organization of the semantic space in different languages and on lexical semantic change.

Acknowledgments

This work has been supported by the French National Research Agency under project ANR-16-CE33-0013. The work is also part of the FoTran project, funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 771113). We thank the anonymous reviewers and the TACL Action Editor for their careful reading of our paper, their thorough reviews, and their helpful suggestions.

References


Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $\&!$*$ vector: Probing sentence embeddings for linguistic properties. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long


Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical


### A Clusterability Metrics

**Variance Ratio.** First, the variance of a cluster $y$ is calculated:

$$\sigma^2(y) = \frac{1}{|y|} \sum_{i \in y} (y_i - \bar{y})^2$$  \hspace{1cm} (3)

where $\bar{y}$ denotes the centroid of cluster $y$. Then the within-cluster variance $W$ and the between-cluster variance $B$ of a clustering solution $C$ are calculated in the following way:

$$W(C) = \sum_{j=1}^{k} p_j \sigma^2(x_j)$$  \hspace{1cm} (4)

$$B(C) = \sum_{j=1}^{k} p_j (\bar{x}_j - \bar{x})^2$$  \hspace{1cm} (5)

where $x$ is the set of all data points and $p_j = \frac{|x_j|}{|x|}$. $x_j$ are the data points in cluster $j$. Finally, the VR of a clustering $C$ is obtained as the ratio between $B(C)$ and $W(C)$:

$$VR = \frac{B(C)}{W(C)}$$  \hspace{1cm} (6)

**Separability (SEP).** In an optimal clustering $C_k$ of the dataset $x$ with $k$ clusters, SEP is defined as follows:

$$SEP(x, k) = \frac{\text{loss}(C_k)}{\text{loss}(C_{k-1})}$$  \hspace{1cm} (7)