What do Toothbrushes do in the Kitchen? How Transformers Think our World is Structured

Alexander Henlein and **Alexander Mehler**
Text Technology Lab, Goethe-University Frankfurt, Germany
{henlein, mehler}@em.uni-frankfurt.de

Abstract

Transformer-based models are now predominant in NLP. They outperform approaches based on static models in many respects. This success has in turn prompted research that reveals a number of biases in the language models generated by transformers. In this paper we utilize this research on biases to investigate to what extent transformer-based language models allow for extracting knowledge about object relations (X *occurs in* Y; X *consists of* Z; *action* A *involves using* X). To this end, we compare contextualized models with their static counterparts. We make this comparison dependent on the application of a number of similarity measures and classifiers. Our results are threefold: Firstly, we show that the models combined with the different similarity measures differ greatly in terms of the amount of knowledge they allow for extracting. Secondly, our results suggest that similarity measures perform much worse than classifier-based approaches. Thirdly, we show that, surprisingly, static models perform almost as well as contextualized models – in some cases even better.

1 Introduction

Few models have recently influenced NLP as much as transformers (Vaswani et al., 2017). Hardly any new NLP system today is introduced without a transformer-based model such as BERT (Devlin et al., 2019) or GPT (Radford et al., 2019). As a result, static models such as word2vec (Mikolov et al., 2013) are increasingly being substituted. Nevertheless, transformers are still far from being fully understood. Thus, research studies are being conducted to find out how they work and what properties the language models they generate have.

During training, transformers seem to capture both syntactic and semantic features (Rogers et al., 2020). For example, dependency trees can be reconstructed from trained attention heads (Clark et al., 2019), syntactic trees can be reconstructed from word encodings (Hewitt and Manning, 2019), and these encodings can be clustered into representations of word senses (Reif et al., 2019). BERT also seems to encode information about entity types and semantic roles (Tenney et al., 2019). For an overview of this research see Rogers et al. (2020).

Since BERT and other transformers are trained on various data crawled from the internet, they are sensitive to biases (Caliskan et al., 2017; May et al., 2019; Bender et al., 2021). In practice, instead of reproducing negative biases, they are expected to allow for the derivation of statements, such as that toothbrushes are spatially associated with bathrooms rather than living rooms. In this line of thinking, approaches such as the popularization of knowledge graphs can be located (Yao et al., 2019; Petroni et al., 2019; Heinzerling and Inui, 2021). Our paper is situated in this context. More specifically, we examine the extent to which knowledge about spatial objects and their relations is implicitly encoded in these models. Since the underlying texts are rather implicit regarding such information, it can be assumed that the object relations derivable from transformers are weakly encoded (cf. Landau and Jackendoff, 1993; Hayward and Tarr, 1995). Reading, for example, the sentence:

"*After getting up, I ate an apple*"

one may assume that the narrator got up from his bed in the bedroom, went to the kitchen, took an apple, washed it in the sink, and finally ate it. The apple is also likely to have been peeled and cut. Equally, however, nothing is said in the sentence about a bedroom or a kitchen. Nevertheless, it is a well known approach to explore the usage regularities of words, currently most efficiently represented by neural networks, as a source for knowledge extraction (see, e.g. Zhang et al., 2017; Bouraoui et al., 2020; Shin et al., 2020; Petroni et al., 2019).

In this work, we use a number of methods to identify biases in contextualized models and ask to what extent they can be used to extract object-

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5791 - 5807

based knowledge from these models. To this end, we consider three relations:

1. *Spatial containment of (source) objects in (target) rooms:* e.g. a fridge probably belongs in a kitchen, but not in a living room;
2. *Parts (source) in relation to composite objects (target):* e.g. a refrigerator compartment is probably a part of a fridge;
3. *Objects (source) in relation to actions (target) that involve them:* e.g. reading involves something being read, e.g., a book.

Regarding these relations, we examine a set of pretrained contextualized and static word representation models. This is done to answer the question to what extent they allow the extraction of instances of these relations when trained on very large datasets. We focus on rather common terms (*kitchen*, *to read* etc.) as part of the general language.

It is assumed that (static or contextualized) models implicitly represent such relations, so that it is possible to identify probable targets starting from certain sources. That is, for a word like *fridge* (source), we expect it to be semantically more strongly associated with *kitchen* (target) than with words naming other rooms, since fridges are more likely to be found in kitchens than in other rooms, and that certain word representation models reflect this association. We also assume that this association is asymmetric and exists to a lesser extent from target to source (cf. Tversky and Gati (2004)).

The paper is organized as follows: Related work is reported in Sec. 2. The datasets we use are represented in Sec. 3 and our method in Sec. 4. Our experiments are presented in Sec. 5 and discussed in Sec. 6. Sec. 7 provides a conclusion. All used data, scripts and results are open source on GitHub[1].

2 Related Work

Biases in NLP models are not a new problem that appeared with BERT, but affect almost all models trained on language datasets (Caliskan et al., 2017). As such, there are methods for measuring social biases in static models such as word2vec (Mikolov et al., 2013). One of the best known approaches is WEAT (Caliskan et al., 2017). Here, two groups of concepts are compared with two groups of attributes based on the difference between the sums of their cosine similarities (see Section 4). This approach already points to a methodological premise that also guides our work: Relations of entities are tentatively determined by similarity analyses of vectorial word representations.

However, a direct comparison of word vectors is not possible with contextualized methods such as BERT, where the vector representation of a word varies with the context of its occurrence (cf. Ethayarajh, 2019). Efforts to transfer the cosine-based approach from static to contextualized models have not been able to recreate similar performances (May et al., 2019). Therefore, new approaches have been developed based on the specifics of contextualized models. For example, BERT is trained using masked language modeling, where the model estimates the probability of masked words in sentences (Devlin et al., 2019). The probability distribution for a masked word in a given context can then be used as information to characterize candidate words (Kurita et al., 2019). Sec. 4.3 describes this approach in more detail. An alternative approach is to examine the interpretability of models (Belinkov and Glass, 2019; Jiang et al., 2020; Petroni et al., 2019, 2020; Bommasani et al., 2020; Hupkes et al., 2020), which goes beyond the scope of this paper. In any event, both approaches share the same basic ideas, e.g., the probability prediction of mask tokens (cf. Kurita et al., 2019; Belinkov and Glass, 2019).

Work has also been done on how BERT represents information about spatial objects. For example, BERT has problems with certain object properties (e.g. *cheap* or *cute*) or implicit visual properties that are rarely expressed (Da and Kasai, 2019). Problems are also encountered with extracting numerical commonsense knowledge, such as the typical number of tires on a car or the feet on a bird (Lin et al., 2020). More than that, the models seem to allow for extracting some object knowledge, but not with respect to properties based on their affordance (e.g. objects through which one can see are transparent (Forbes et al., 2019)). Even though these results seem to question the use of BERT and its competitors for knowledge extraction, these models still perform better in downstream tasks than their static competitors (Devlin et al., 2019; Liu et al., 2019; Brown et al., 2020; Da and Kasai, 2019). Bouraoui et al. (2020) compared these models using different datasets and lexical relations. These include relations similar to those examined here (e.g. a pot is usually found in a kitchen), but beyond the level of detail achieved in

[1] https://github.com/texttechnologylab/SpatialAssociationsInLM

our study.

What will become increasingly important is the so-called grounding of language models (Merrill et al., 2021): Here, the models are trained not only on increasingly large text data, but also, for example, on images thus enabling better "understanding" of spatial relations (Sileo, 2021; Li et al., 2020). In this paper, we focus on models without grounding.

3 Datasets Used for Evaluation

3.1 Spatial Containment

The *NYU Depth V2 Dataset* (Silberman et al., 2012) consists of video sequences of numerous indoor scenes. It features 464 labeled scenes using a rich category set. We use this dataset as a basis for evaluating the probability of occurrence of objects in rooms (e.g. kitchen, living room, etc.). That is, we estimate the conditional probability $P(r \mid o)$ of a room r (target) given an object o (source). In this way, we aim to measure the strength of an object's association with a particular room as reflecting the corresponding spatial containment relation. At the same time, we want to filter out objects such as *window* that are evenly distributed among the rooms studied here. In our experiments, we consider the ten most frequently mentioned objects in NYU to associate with the five most frequently mentioned spaces. This data is shown in the Table 4 (appendix).

The advantage of NYU over other scene datasets such as 3D-Front (Fu et al., 2020) is that it deals with real spaces and not artificially created ones. In addition, NYU's object category set is relatively fine-grained (we counted 895 different object names) and uses colloquial terms. This is in contrast to, for example, SUNCG (Song et al., 2017) (with categories like "slot machine with chair","range hood with cabinet", "food processor") and ShapeNetCore (Chang et al., 2015) with only 55 object categories or COCO (Lin et al., 2014) with 80 object categories. This makes NYU more suitable for our task of evaluating word representation models as resources for knowledge extraction starting from general language.

3.2 Part-whole Relations

We use a subset of the object descriptions from *Online-Bildwörterbuch*[2]. This resource describes very fine-grained part-whole relations of objects expressed by colloquial names, in contrast to, e.g., PartNet (Mo et al., 2019) where one finds labels such as *seat single surface* or *arm near vertical bar*. The list of objects from *Online-Bildwörterbuch* used in our study and their subdivisions is shown in Table 5. The selected objects were chosen by hand, provided that the chosen examples are general enough and the subdivision is sufficiently fine.

3.3 Action-object Relations

To study entities as typical objects of certain actions, we derive a dataset from HowToKB (Chu et al., 2017) which is based on WikiHow[3]. In HowToKB, task frames, temporal sequences of subtasks, and attributes for involved objects were extracted from WikiHow articles. Some changes were made to the knowledge database, including a newly crawled version of WikiHow. In addition, the pre-processing tools have been updated and partially extended (see Table 6).

3.3.1 Related Datasets

For evaluating static models, there are datasets and approaches to measuring lexical relations, such as DiffVec (Vylomova et al., 2016), BATS (Gladkova et al., 2016) or BLiMP (Warstadt et al., 2020). Although these datasets are also used to evaluate BERT (Bouraoui et al., 2020), they represent only an unstructured subset of the data we used and are thus not appropriate for our study.

4 Approach

We now present the static and contextualized models used in our study. Table 7 in the appendix lists these models and their sources. We also specify the measures used to compute word associations as a source of knowledge extraction, and describe how to use classifiers as an alternative to them.

4.1 Static Models

Probably the best known static model is word2vec (Mikolov et al., 2013). Its CBOW variant is trained to predict words in the context of their surrounding words. The word representations trained in this way partially encode semantic relations (Mikolov et al., 2013), making them a suitable candidate for comparison with the corresponding information values of contextualized word representations. In addition to word2vec, we consider GloVe (Pennington et al., 2014), Levy (Levy and Goldberg, 2014),

[2] http://www.bildwoerterbuch.com/en/home

[3] https://www.wikihow.com/

fastText (Mikolov et al., 2018) and a static BERT model (Gupta and Jaggi, 2021). Unlike window-based approaches to static embeddings, Levy embeddings are trained on dependency trees.

4.2 Contextualized Models

Unlike static models, the vector representations of (sub-)word (units) in contextualized models depend on the context in which they occur so that tokens of the same type may each be represented differently in different contexts. All contextual models we evaluate here are pre-trained and come from the *huggingface models repository*[4]. We evaluate two types of contextualized models:

Masked Language Models (MLM) are trained to reconstruct randomly masked words in input sequences. We experiment with BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020) and ALBERT (Lan et al., 2019). The models differ in training, training data, and model size. BERT is trained using masked language modeling and next sentence prediction. RoBERTa omits the second task, but uses much more training data. Two models are trained for ELECTRA: one on masked language modeling (generator) and a second one that recognizes just these replaced tokens (discriminator). Since many of our evaluations need mask tokens, we only use the generator model for the evaluations. Finally, ALBERT is trained to predict the order of pairs of consecutive text segments in addition to masked language modeling.

Causal Language Models (CLM) are trained to predict the next word for a given input text. From this class we experiment with GPT-2 (Radford et al., 2019), GPT-Neo (Gao et al., 2021; Black et al., 2021) and GPT-J (Wang and Komatsuzaki, 2021). GPT-Neo and GPT-J are re-implementations of GPT-3 (Brown et al., 2020) where GPT-J was trained on a significantly larger data set named *The Pile* (Gao et al., 2021) (cf. Table 7 in the appendix).

4.3 Similarity Measures

To compute similarities of word associations based on the models studied here, we make use of research on biases in such models. These approaches calculate biases between two groups of concepts with respect to candidate groups of attributes. To this end, associations are evaluated by computing the similarities of vector representations of concepts and attributes. We adopt this approach to investigate our research question. However, as we consider knowledge extraction starting from source words (e.g. *toaster*, *shower*) in relation to target words (e.g. *kitchen*, *bathroom*), we modify it as described below.

4.3.1 Cosine and Correlation Measures

Based on the human implicit association test (Greenwald et al., 1998), WEAT (Caliskan et al., 2017) is originally designed to compare the association between two sets of concepts (X and Y) and two sets of attributes (A and B). The degree of bias is calculated as follows:

$$s(X, Y, A, B) = \sum_{x \in X} s(x, A, B) - \sum_{y \in Y} s(y, A, B) \quad (1)$$

$$s(w, A, B) = \sum_{a \in A} \cos(w, a) - \sum_{b \in B} \cos(w, b) \quad (2)$$

Since we are considering source words in relation to target words, we use the following variant:

$$s(X, A) = \frac{1}{|X||A|} \sum_{x \in X} \sum_{a \in A} \cos(x, a) \quad (3)$$

For contextualized models, we adopt the approach of May et al. (2019), that is, we generate sentences such as "This is a {x}." or "A {x} is here". All templates used in our study are listed in the appendix Table 8. However, instead of using the BERT token [CLS] (the default token at the beginning of an input sequence, which often serves as the default representation of the entire sequence), we use the maximum of the vector representations of all subwords of the expression. This approach is suitable for models like RoBERTa that do not use the [CLS] token for training, or the GPT models that do not have this token at all. In addition, we also achieved slightly better results on regular BERT models using this approach. We explain this with the fact that our focus is actually only on single tokens and that the vector representation of the [CLS] token often focuses only on a few dimensions (Zhou et al., 2019). Our approach results in a set of contextualized representations for each source and target word, which are then compared using formula 3. We were able to obtain better results in our experiments with this representation than with those generated via the [CLS] token. For static models, if there is no vector representation

[4] https://huggingface.co/models

for a potential multiword expressions (MWE)[5], the average of the vectors of their components is used. This representation yielded the largest bias in the work of Azarpanah and Farhadloo (2021). For the static models, we also experimented with *distance correlation* (Székely et al., 2007), *Pearson correlation* (Benesty et al., 2009), *Spearman correlation* (Kokoska and Zwillinger, 2000), *Kendall's tau* (Kendall, 1938) and *Mahalanobis distance* (Mahalanobis, 1936) – cf. Torregrossa et al. (2020); Azarpanah and Farhadloo (2021) – of the word vectors. Due to space limitations, only the values of the distance correlation and Kendall's tau are shown (see Table 1); the other correlation measures behave similarly. Moreover, the values for these measures tend to perform worse for contextualized models. This observation is consistent with findings of Azarpanah and Farhadloo (2021) where the Mahalanobis distance measure performed worst.

4.3.2 Increased Log Probability

The cosine measure has shown to be problematic for assessing bias in contextualized models such as BERT (May et al., 2019; Kurita et al., 2019). Kurita et al. (2019) have therefore developed a new approach for models trained using masked language modeling. They weight the probability of a target word in a simple sentence template, assuming that an attribute is given or not:

$$\text{score}(target, attribute) = \log \frac{P([\text{MASK}] = [target] \mid [\text{MASK}]\ is\ a\ [attribute])}{P([\text{MASK}_1] = [target] \mid [\text{MASK}_1]\ is\ a\ [\text{MASK}_2])}$$

Experiments show that the values of this measure correlate significantly better with human biases.

Since this measure is based on the context sensitivity of models, it cannot be applied to static models. For contextualized models, we use the probability of the last token (e.g. *curtain* in the case of *shower curtain*) for source-forming MWEs and the first token (e.g. *living* in the case of *living room*) for target-forming MWEs. We also performed experiments with multiple masks, one for each of the components of a MWE. However, this did not produce better results. We adapt this approach for causal language models as follows: Instead of a complete sentence, we use incomplete sentence templates such as "A(n) {object} is usually in the ..." or "In the {room} is usually a/an ...". The model should then predict the next token. Instead of masking the seed word, a neutral equivalent is used for calculation:

A(n) {object} is usually in the ...
⇓
This is usually in the ....

Instead of performing the analysis in only one direction, we determine the score for both the target and the source given the other.

4.3.3 Classifier-based Measures

In addition to the previously described measures, we experiment with classifiers. To this end, we train three classifiers on the model representations of the source word to determine the associated target word as a class label (e.g. predict *kitchen*, given the vector of *frying pan*). We generate the set of source word representations X in the same way as in the case of the cosine measure (see Section 4.3.1) and average them before classification:

$$target = Classifier\left(\frac{1}{|X|}\sum_{\vec{x}\in X}\vec{x}\right)$$

The training runs on a leave-one-out cross-validation repeated 100 times. The target vector was then generated from the counted predicted classes (see Figure 2b in Appendix). We trained a k-nearest neighbors classifier with $k = 5$ (KNN), an SVM with a linear kernel and a feed-forward network (FFN). A small hyperparameter optimization was performed for the FFN, which resulted in the following parameters: Adam Optimizer (Kingma and Ba, 2014) with a learning rate of 0.01 over 100 epochs and one hidden layer of size 100 and ReLU as activation function.

4.4 Scoring Measures and Classifiers

Given a word representation model, we compute the final score for the measures and classifiers to estimate how well they reconstruct the original probability distribution of the source entities relative to the target entities (see Table 4, 5, and 6). This is computed by the distance correlation (Székely et al., 2007) between the target-source probability distributions and the corresponding association distributions of the respective measure or classifier. The advantage of the distance correlation over the Pearson correlation is that it can also measure nonlinear relations. This was calculated both for all targets individually (correlation of all sources to one target) and then *concatenated* for all targets

[5] Word2Vec contains vectors for MWE's.

together; we denote this variant by *CONC*. Therefore, *CONC* does not correspond to the average of the individual distance correlations.

5 Experiments

Using the apparatus of Section 4, we now evaluate the classes of word representation models (static, MLMs and CLMs) in conjunction with the similarity measures and classifiers. The results for the static models are shown in Table 1, for the MLMs in Table 2 and for the CLMs in Table 3. Figure 2, 3 and 4 in Appendix show a visualization of the associations computed by means of cosine, masked-target & masked-source increased log similarity measures and the FFN classifier based on BERT-Large using the different datasets.

An experiment was also conducted with word frequencies via *Google Ngram*[6] (see Section A.1 in the appendix).

5.1 Model-related Observations

The basic expectation that the cosine measure would generally perform the worst and the FFN classifier the best was met (see Tables 1–3). Interestingly, cosine is also outperformed by distance correlation in almost all cases.

Among the static models, GloVe and fastText performed best in most cases, especially on the room and part dataset (Table 1). Although Levy performs by far the worst in the room dataset, it keeps up with all classification results in the verb dataset. One reason for this could be the dependency-based learning strategy, which seems to work very well for verb associations, even though it was trained on a much smaller data set.

Among the masked-language models, BERT-Base surprisingly performed the best (Table 2). BERT-Large achieved the better Increased Log Probabilities, but the FFN classifier still worked better with the vector representations of the Base variant. This suggests that although associations are represented in a more fine-grained manner in BERT-Large, they are more difficult to retrieve due to the size of this model.

Among the masked-language models, GPT-J (which was trained with by far the largest training data) performs best (Table 3). Context-based models generally seem to determine the target given the source ($P(\textit{target} \mid \textit{source})$) more easily than the reverse ($P(\textit{source} \mid \textit{target})$). With verbs, on the other hand, the reverse effect occurs. The GPT models show that the results for sources are better when weighted, while for targets the results are better without weighting.

In general, the SVM performed surprisingly well, even though only a linear kernel was used. But also the KNN method mostly performed better than the similarity measures. However, FFN performs best in all cases, outperforming cosine (worst case) by increases in the interval $[6\%, 52\%]$ and outperforming the KNN approach (worst classifier) in each case by increases in the interval $[2\%, 43\%]$.

5.2 Dataset-related Observations

In terms of rooms, *bathroom* scores the best, while *living room* or *office* usually score the worst. This may be because many bathroom objects are related to specific bathroom activities (e.g., toothbrush, bathtub), while objects that used to be located in the living room are increasingly found in other rooms (e.g., television in the bedroom). This would also explain why the results for *kitchen* are also better.

On the part dataset, the static models actually performed significantly better than the contextualized models. This relates especially to GloVe and fastText which outperformed almost all contextualized models. Thus, static models are in some cases a good alternative to their contextualized counterparts. However, the more technical the objects become (here *mortise lock* and *dishwasher*), the worse the results become.

On the verb dataset, the contextualized models perform minimally better. As mentioned earlier, these models can associate objects with verbs more easily than the other way around. Here, the largest difference in performance is observed in the case of Levy, where the results are almost equal to those of the other models, probably due to the learning strategy based on dependency trees.

In summary, knowledge extraction using language models, whether static or contextualized, is more effective using classifiers than using similarity measures commonly used in the field of bias research: there is potential for this type of knowledge extraction, but at the price of training classifiers – if one uses similarity measures instead, this knowledge is mostly out of reach.

5.3 Relation Observation

All previous evaluations only examined associations between instances and concepts, but not whether the models represent their true relations.

[6] https://books.google.com/ngrams

		Word2Vec						GloVe						Levy						fastText						static-BERT					
		cos	dist	kend	knn	svm	ffn	cos	dist	kend	knn	svm	ffn	cos	dist	kend	knn	svm	ffn	cos	dist	kend	knn	svm	ffn	cos	dist	kend	knn	svm	ffn
Room	bathroom	0.37	0.37	0.37	0.39	0.62	0.82	0.38	0.39	0.38	0.57	0.93	0.93	0.39	0.40	0.39	0.14	0.34	0.37	0.53	0.53	0.52	0.73	0.67	0.90	0.54	0.50	0.50	0.25	0.66	0.70
	bedroom	0.20	0.20	0.20	0.13	0.49	0.70	0.31	0.29	0.30	0.28	0.66	0.45	0.21	0.21	0.21	0.10	0.25	0.11	0.30	0.31	0.32	0.26	0.44	0.59	0.28	0.27	0.27	0.35	0.33	0.35
	kitchen	0.35	0.34	0.35	0.20	0.55	0.53	0.37	0.40	0.41	0.52	0.65	0.81	0.17	0.17	0.18	0.09	0.32	0.30	0.38	0.36	0.34	0.41	0.66	0.76	0.40	0.41	0.41	0.45	0.53	0.68
	living room	0.23	0.23	0.24	0.06	0.33	0.35	0.30	0.27	0.28	0.10	0.49	0.51	0.24	0.24	0.23	0.40	0.16	0.25	0.25	0.26	0.24	0.09	0.36	0.60	0.19	0.19	0.19	0.00	0.10	0.46
	office	0.28	0.28	0.26	0.51	0.51	0.55	0.14	0.31	0.35	0.51	0.59	0.64	0.25	0.27	0.28	0.40	0.36	0.25	0.25	0.30	0.33	0.45	0.32	0.63	0.40	0.44	0.45	0.10	0.21	0.32
	CONC	**0.23**	**0.23**	**0.23**	**0.22**	**0.50**	**0.60**	**0.27**	**0.31**	**0.32**	**0.37**	**0.67**	**0.67**	**0.16**	**0.15**	**0.15**	**0.15**	**0.11**	**0.23**	**0.30**	**0.31**	**0.31**	**0.40**	**0.45**	**0.70**	**0.31**	**0.31**	**0.31**	**0.18**	**0.39**	**0.48**
Part	bed	0.41	0.41	0.40	0.64	0.56	0.56	0.38	0.51	0.51	0.56	0.76	0.84	-	-	-	-	-	-	0.42	0.51	0.52	0.69	0.61	0.67	0.47	0.48	0.46	0.16	0.59	0.54
	dishwasher	0.19	0.23	0.23	0.06	0.37	0.27	0.33	0.32	0.30	0.03	0.19	0.32	-	-	-	-	-	-	0.35	0.33	0.33	0.06	0.13	0.23	0.17	0.17	0.17	0.13	0.28	0.31
	door	0.12	0.11	0.11	0.54	0.75	0.75	0.19	0.23	0.22	0.48	0.81	0.85	-	-	-	-	-	-	0.25	0.27	0.24	0.36	0.55	0.84	0.24	0.25	0.25	0.36	0.73	0.67
	mortise lock	0.15	0.16	0.16	0.16	0.50	0.54	0.22	0.26	0.28	0.45	0.74	0.68	-	-	-	-	-	-	0.11	0.17	0.20	0.68	0.55	0.68	0.20	0.21	0.21	0.14	0.49	0.47
	refrigerator	0.44	0.46	0.46	0.51	0.47	0.52	0.53	0.57	0.56	0.55	0.55	0.66	-	-	-	-	-	-	0.54	0.58	0.58	0.28	0.40	0.55	0.50	0.50	0.50	0.56	0.56	0.53
	toilet	0.28	0.28	0.28	0.01	0.49	0.55	0.33	0.33	0.32	0.31	0.63	0.60	-	-	-	-	-	-	0.37	0.34	0.33	0.55	0.50	0.72	0.24	0.23	0.23	0.34	0.57	0.58
	CONC	**0.25**	**0.27**	**0.26**	**0.28**	**0.52**	**0.53**	**0.30**	**0.34**	**0.34**	**0.39**	**0.60**	**0.65**	**-**	**-**	**-**	**-**	**-**	**-**	**0.28**	**0.33**	**0.33**	**0.35**	**0.43**	**0.61**	**0.29**	**0.29**	**0.29**	**0.23**	**0.54**	**0.52**
Verb	eat	0.79	0.79	0.77	0.89	0.89	0.89	0.77	0.86	0.80	0.89	0.89	0.92	0.46	0.45	0.45	0.66	0.87	0.87	0.73	0.80	0.79	0.69	0.89	0.89	0.83	0.84	0.83	0.61	0.89	0.87*
	listen to	0.54	0.64	0.56	0.21	0.38	0.46	0.59	0.70	0.65	0.06	0.53	0.49	0.28	0.22	0.23	0.20	0.38	0.52	0.42	0.53	0.63	0.21	0.42	0.40	0.54	0.56	0.53	0.00	0.39	0.50
	play	0.64	0.69	0.64	0.60	0.66	0.60	0.65	0.80	0.73	0.43	0.45	0.45	0.44	0.45	0.43	0.41	0.50	0.57	0.63	0.69	0.68	0.28	0.66	0.66	0.56	0.56	0.54	0.00	0.49	0.63*
	read	0.43	0.52	0.48	0.38	0.59	0.61	0.51	0.60	0.59	0.48	0.53	0.50	0.31	0.31	0.31	0.49	0.31	0.50	0.54	0.56	0.59	0.42	0.50	0.59	0.48	0.52	0.48	0.00	0.31	0.47
	wash with	0.53	0.54	0.53	0.48	0.61	0.63	0.48	0.57	0.53	0.66	0.66	0.62	0.37	0.34	0.35	0.41	0.66	0.62	0.45	0.51	0.49	0.67	0.66	0.66	0.39	0.40	0.40	0.11	0.55	0.61
	wear	0.76	0.78	0.76	0.88	0.84	0.88	0.80	0.87	0.84	0.88	0.83	0.85	0.56	0.52	0.50	0.82	0.85	0.85	0.77	0.80	0.79	0.59	0.93	0.92	0.78	0.82	0.80	0.72	0.81	0.84
	CONC	**0.58**	**0.60**	**0.57**	**0.56**	**0.64**	**0.67**	**0.59**	**0.68**	**0.65**	**0.55**	**0.65**	**0.65**	**0.34**	**0.32**	**0.31**	**0.46**	**0.59**	**0.65**	**0.51**	**0.58**	**0.58**	**0.43**	**0.66**	**0.68**	**0.54**	**0.55**	**0.54**	**0.15**	**0.56**	**0.65**

Table 1: All results of the static models. cos: Cosine Measure, dist: Distance Correlation, kend: Kendall's Tau, knn: K-Nearest Neighbors, svm: Support Vector Machine, fnn: Feed-Forward Network. The gap in Levy is due to its small training set and the corresponding small vocabulary. (A *gray cell* indicates significant values at $p < 0.01$)

		BERT-Base						BERT-Large						RoBERTa						ElectraGen						Albert					
		cos	m-s	m-t	knn	svm	ffn	cos	m-s	m-t	knn	svm	ffn	cos	m-s	m-t	knn	svm	ffn	cos	m-s	m-t	knn	svm	ffn	cos	m-s	m-t	knn	svm	ffn
Room	bathroom	0.57	0.13	0.52	0.72	0.87	0.93	0.65	0.30	0.59	0.78	0.93	0.93	0.21	0.24	0.52	0.55	0.83	0.88	0.58	0.32	0.34	0.49	0.72	0.73	0.24	0.18	0.39	0.52	0.75	0.90
	bedroom	0.48	0.33	0.43	0.53	0.66	0.77	0.44	0.41	0.44	0.44	0.87	0.78	0.23	0.18	0.36	0.17	0.53	0.60	0.32	0.31	0.37	0.37	0.37	0.39	0.23	0.22	0.47	0.31	0.44	0.68
	kitchen	0.56	0.25	0.58	0.62	0.81	0.83	0.43	0.24	0.54	0.72	0.77	0.79	0.39	0.27	0.59	0.16	0.62	0.73	0.34	0.24	0.36	0.48	0.34	0.39	0.25	0.17	0.30	0.05	0.56	0.69
	living room	0.30	0.37	0.26	0.51	0.78	0.79	0.23	0.38	0.24	0.57	0.49	0.66	0.13	0.38	0.28	0.49	0.74	0.65	0.26	0.48	0.33	0.15	0.27	0.26	0.15	0.35	0.54	0.20	0.29	0.40
	office	0.46	0.39	0.28	0.40	0.59	0.61	0.40	0.37	0.31	0.25	0.52	0.71	0.14	0.37	0.38	0.18	0.74	0.63	0.17	0.37	0.23	0.42	0.27	0.36	0.23	0.22	0.42	0.45	0.66	0.81
	CONC	**0.43**	**0.26**	**0.33**	**0.54**	**0.73**	**0.78**	**0.34**	**0.26**	**0.36**	**0.55**	**0.72**	**0.78**	**0.19**	**0.22**	**0.31**	**0.28**	**0.69**	**0.71**	**0.22**	**0.30**	**0.27**	**0.38**	**0.40**	**0.43**	**0.19**	**0.15**	**0.23**	**0.25**	**0.53**	**0.69**
Part	bed	0.55	0.41	0.51	0.51	0.69	0.79	0.49	0.41	0.55	0.56	0.69	0.69	0.20	0.42	0.62	0.49	0.52	0.60	0.37	0.31	0.43	0.44	0.44	0.43	0.26	0.40	0.54	0.36	0.66	0.71
	dishwasher	0.22	0.16	0.22	0.27	0.31	0.28	0.30	0.18	0.31	0.29	0.17	0.18	0.16	0.19	0.19	0.13	0.24	0.17	0.26	0.19	0.21	0.01	0.23	0.36	0.17	0.18	0.25	0.26	0.25	0.23
	door	0.19	0.32	0.20	0.34	0.65	0.63	0.13	0.28	0.39	0.47	0.60	0.62	0.15	0.33	0.27	0.52	0.42	0.51	0.14	0.20	0.17	0.41	0.57	0.60	0.13	0.29	0.21	0.36	0.50	0.54
	mortise lock	0.12	0.14	0.09	0.16	0.26	0.28	0.14	0.23	0.11	0.19	0.26	0.35	0.07	0.29	0.12	0.08	0.18	0.28	0.16	0.18	0.15	0.39	0.59	0.39	0.09	0.27	0.22	0.16	0.31	0.39
	refrigerator	0.44	0.21	0.40	0.48	0.47	0.54	0.38	0.21	0.54	0.42	0.51	0.50	0.18	0.38	0.45	0.49	0.43	0.49	0.37	0.33	0.43	0.46	0.45	0.53	0.44	0.27	0.51	0.66	0.51	0.61
	toilet	0.18	0.16	0.29	0.16	0.34	0.45	0.25	0.16	0.26	0.36	0.55	0.50	0.22	0.34	0.41	0.45	0.51	0.51	0.34	0.26	0.42	0.26	0.41	0.46	0.24	0.23	0.25	0.22	0.31	0.46
	CONC	**0.20**	**0.20**	**0.24**	**0.33**	**0.45**	**0.49**	**0.22**	**0.21**	**0.28**	**0.39**	**0.46**	**0.46**	**0.07**	**0.29**	**0.29**	**0.39**	**0.39**	**0.43**	**0.21**	**0.19**	**0.23**	**0.32**	**0.45**	**0.47**	**0.08**	**0.23**	**0.27**	**0.35**	**0.42**	**0.49**
Verb	eat	0.78	0.65	0.67	0.89	0.84	0.90	0.65	0.58	0.72	0.80	0.89	0.90	0.26	0.66	0.81	0.65	0.87	0.86	0.62	0.64	0.76	0.74	0.79	0.79	0.53	0.61	0.74	0.57	0.84	0.85
	listen to	0.46	0.53	0.51	0.42	0.52	0.57	0.50	0.52	0.50	0.43	0.55	0.52	0.30	0.53	0.55	0.23	0.49	0.54	0.57	0.47	0.59	0.00	0.36	0.39	0.23	0.47	0.51	0.07	0.44	0.57
	play	0.63	0.58	0.69	0.54	0.58	0.61	0.55	0.60	0.73	0.54	0.64	0.66	0.37	0.64	0.65	0.38	0.53	0.59	0.64	0.53	0.69	0.64	0.64	0.65	0.37	0.42	0.52	0.45	0.60	0.62
	read	0.42	0.46	0.65	0.34	0.73	0.65	0.30	0.42	0.66	0.42	0.77	0.59	0.26	0.29	0.59	0.21	0.44	0.44	0.41	0.43	0.63	0.51	0.68	0.69	0.31	0.19	0.57	0.35	0.63	0.60
	wash with	0.49	0.46	0.33	0.49	0.66	0.63	0.42	0.53	0.45	0.61	0.62	0.60	0.30	0.56	0.30	0.23	0.60	0.59	0.42	0.50	0.35	0.52	0.40	0.41	0.33	0.42	0.32	0.18	0.46	0.51
	wear	0.66	0.64	0.76	0.88	0.90	0.92	0.62	0.57	0.74	0.79	0.90	0.85	0.24	0.64	0.77	0.36	0.72	0.79	0.53	0.62	0.74	0.90	0.84	0.83	0.30	0.61	0.77	0.61	0.77	0.86
	CONC	**0.53**	**0.53**	**0.38**	**0.59**	**0.69**	**0.71**	**0.37**	**0.50**	**0.44**	**0.60**	**0.73**	**0.68**	**0.20**	**0.55**	**0.37**	**0.28**	**0.59**	**0.64**	**0.49**	**0.51**	**0.37**	**0.60**	**0.61**	**0.62**	**0.15**	**0.40**	**0.26**	**0.29**	**0.62**	**0.67**

Table 2: All results of the contextual masked-language models. cos: Cosine Measure, m-s: Masked-Source Log Score, m-t: Masked-Target Log Score, knn: K-Nearest Neighbors, svm: Support Vector Machine, fnn: Feed-Forward Network. (A *gray cell* indicates significant values at $p < 0.01$)

		GPT2								GPT-Neo								GPT-J							
		cos	p-s	p-s-l	p-t	p-t-l	knn	svm	ffn	cos	p-s	p-s-l	p-t	p-t-l	knn	svm	ffn	cos	p-s	p-s-l	p-t	p-t-l	knn	svm	ffn
Room	bathroom	0.52	0.20	0.38	0.50	0.37	0.31	0.95	0.95	0.30	0.22	0.51	0.36	0.25	0.53	0.89	0.91	0.50	0.26	0.60	0.66	0.48	0.35	0.89	0.92
	bedroom	0.26	0.31	0.23	0.47	0.38	0.26	0.61	0.54	0.19	0.33	0.21	0.53	0.48	0.55	0.49	0.57	0.24	0.32	0.23	0.62	0.48	0.33	0.70	0.64
	kitchen	0.34	0.41	0.45	0.69	0.60	0.53	0.82	0.83	0.31	0.49	0.67	0.70	0.57	0.38	0.51	0.81	0.33	0.36	0.52	0.83	0.70	0.70	0.82	0.83
	living room	0.21	0.43	0.26	0.41	0.33	0.16	0.27	0.46	0.26	0.57	0.39	0.60	0.44	0.28	0.13	0.48	0.21	0.50	0.47	0.67	0.45	0.46	0.40	0.63
	office	0.13	0.21	0.43	0.37	0.23	0.31	0.44	0.73	0.33	0.30	0.43	0.46	0.34	0.24	0.53	0.72	0.23	0.36	0.53	0.49	0.39	0.37	0.52	0.69
	CONC	**0.26**	**0.23**	**0.30**	**0.46**	**0.35**	**0.30**	**0.61**	**0.72**	**0.15**	**0.34**	**0.44**	**0.44**	**0.35**	**0.40**	**0.52**	**0.71**	**0.23**	**0.32**	**0.42**	**0.56**	**0.41**	**0.42**	**0.66**	**0.74**
Part	bed	0.36	0.30	0.51	0.67	0.45	0.55	0.59	0.70	0.32	0.38	0.46	0.77	0.70	0.66	0.78	0.88	0.46	0.38	0.36	0.81	0.68	0.71	0.83	0.84
	dishwasher	0.11	0.34	0.22	0.25	0.23	0.18	0.21	0.28	0.06	0.23	0.30	0.30	0.29	0.15	0.15	0.24	0.09	0.26	0.30	0.44	0.38	0.12	0.15	0.32
	door	0.23	0.07	0.14	0.20	0.28	0.20	0.65	0.66	0.27	0.10	0.17	0.35	0.42	0.20	0.44	0.66	0.15	0.13	0.12	0.37	0.41	0.25	0.67	0.77
	mortise lock	0.07	0.34	0.43	0.17	0.18	0.27	0.63	0.65	0.11	0.49	0.42	0.30	0.22	0.27	0.49	0.61	0.15	0.43	0.43	0.47	0.31	0.04	0.63	0.66
	refrigerator	0.42	0.41	0.24	0.53	0.52	0.47	0.39	0.51	0.29	0.33	0.33	0.47	0.55	0.44	0.47	0.57	0.46	0.51	0.41	0.57	0.63	0.55	0.53	0.63
	toilet	0.29	0.36	0.44	0.20	0.17	0.16	0.49	0.54	0.27	0.42	0.50	0.25	0.26	0.23	0.50	0.58	0.32	0.45	0.48	0.32	0.37	0.26	0.53	0.62
	CONC	**0.14**	**0.24**	**0.28**	**0.28**	**0.25**	**0.29**	**0.47**	**0.54**	**0.12**	**0.30**	**0.34**	**0.37**	**0.36**	**0.32**	**0.46**	**0.57**	**0.16**	**0.34**	**0.32**	**0.42**	**0.43**	**0.33**	**0.54**	**0.62**
Verb	eat	0.49	0.82	0.65	-	-	0.82	0.87	0.87	0.45	0.86	0.66	-	-	0.76	0.87	0.88	0.48	0.68	0.74	-	-	0.63	0.89	0.89
	listen to	0.22	0.57	0.51	-	-	0.29	0.50	0.58	0.22	0.47	0.47	-	-	0.20	0.42	0.55	0.21	0.60	0.56	-	-	0.29	0.52	0.60
	play	0.40	0.64	0.62	-	-	0.62	0.61	0.59	0.32	0.66	0.61	-	-	0.20	0.55	0.62	0.34	0.66	0.70	-	-	0.37	0.67	0.67
	read	0.32	0.63	0.30	-	-	0.32	0.45	0.45	0.32	0.61	0.34	-	-	0.29	0.52	0.49	0.40	0.63	0.41	-	-	0.20	0.59	0.49
	Wash with	0.39	0.77	0.51	-	-	0.57	0.61	0.63	0.28	0.66	0.52	-	-	0.39	0.41	0.60	0.23	0.69	0.52	-	-	0.66	0.61	0.64
	wear	0.44	0.38	0.72	-	-	0.76	0.84	0.87	0.16	0.39	0.66	-	-	0.68	0.79	0.85	0.21	0.38	0.62	-	-	0.87	0.84	0.87
	CONC	**0.31**	**0.52**	**0.53**	**-**	**-**	**0.51**	**0.64**	**0.66**	**0.19**	**0.49**	**0.52**	**-**	**-**	**0.40**	**0.59**	**0.66**	**0.23**	**0.50**	**0.56**	**-**	**-**	**0.48**	**0.68**	**0.69**

Table 3: All results of the contextual causal-language models. p-s: Predict Source Score, p-s-l: : Predict Source Log Score, p-t: Predict Target Score, p-t-l: Predict Target Log Score. The gap for the verb p-t score is due to the lack of an easily applicable sentence templates in this direction. (A *gray cell* indicates significant values at $p < 0.01$)

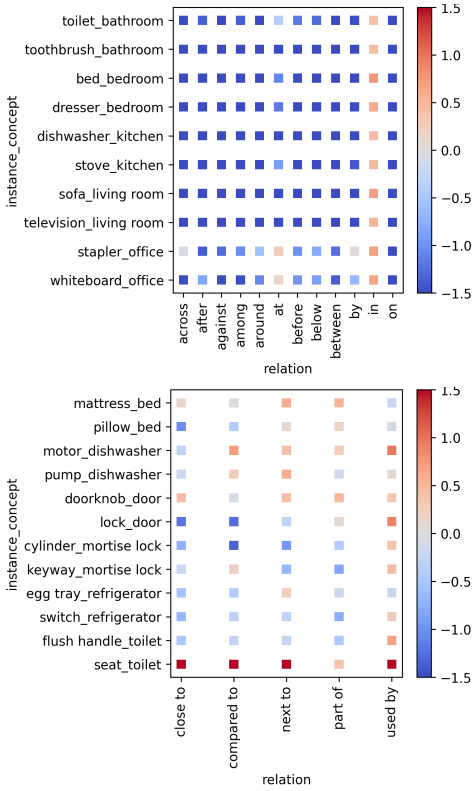


Figure 1: Small relation evaluation of BERT-large after the method of Kurita et al. (2019).

To fill this gap, we repeated the experimental setup of Kurita et al. (2019) for the room and part dataset on BERT-large, but this time masked the relation. The results are shown in Figure 1. Our selection of relations does not claim to be exhaustive, but serves as an illustration. It shows that while BERT-large is still very good at assigning objects *in* rooms, the dominant relation predicted for parts is *used by*. This suggests that BERT has problems correctly assigning object parts, an observation that could explain its poorer results while being consistent with findings of (Lin et al., 2020) (e.g., regarding counting parts).

6 Discussion

As good as the results obtained using classifiers are, they must be viewed with caution. One can attribute their success to the fine-tuning of numerous parameters (and ultimately to overfitting); however, one can also attribute this success to nonlinear structuring of the information encoded in language models. In other words, these models appear to encode object knowledge, but require a sophisticated apparatus to retrieve it. Thus, they should not be considered as an alternative to unsupervised approaches.

Another issue is that our experiments do not yet allow for a comparison of model *architectures*, as the models studied differ significantly in terms of the size of their parameter spaces and training data. Our experiments do suggest that certain smaller models come close to or even outperform the results of larger models. However, a comparison of model architectures would require controlling for these parameters. Nevertheless, the results we have obtained are, in part, promising enough to encourage such research.

Finally, our experiments show that static models can perform better than contextualized models to some extent. This finding is conditioned by our experiments and their context of application. These observations that *older* models perform better on certain tasks are consistent with other work (e.g. LSTMs on small datasets for intent classification (Ezen-Can, 2020) or definiteness prediction (Kabbara and Cheung, 2021). At this point, a much broader analysis is needed (considering more areas and object relations), which also exploits contextual knowledge represented in contextualized models more than has been done here and in related work. Nevertheless, it is generally difficult to obtain data for such a broader analysis, and our experiments are already broader in scope and consider finer relationships than similar approaches.

7 Conclusion

We evaluated static and contextualized models as potential resources for object-related knowledge extraction. To this end, we examined three datasets (to identify typical artifacts in rooms, objects of actions, or parts of objects). We also experimented with different similarity measures and classifiers to extract the information contained in the language models. In doing so, we have shown that the models in combination with the measures differ greatly in terms of the amount of knowledge they allow for extracting. There is a weak trend that BERT-Base is the best performer among contextualized models, and GloVe and fastText among static models. Secondly, our results suggest that approaches based on classifiers perform significantly better than similarity measures. Thirdly, we have shown that static models perform almost as well as contextualized models – in some cases even better. This result shows that research on these models needs to be advanced. In future work we will also investigate

how grounded language models perform on such datasets. However, as noted above, this requires a significant expansion of bias research, such as that conducted here to enable knowledge extraction.

Acknowledgements

The support by the *Stiftung Polytechnische Gesellschaft* (SPTG) is gratefully acknowledged.

References

Hossein Azarpanah and Mohsen Farhadloo. 2021. Measuring biases of word embeddings: What similarity measures and descriptive statistics to use? In *Proceedings of the First Workshop on Trustworthy Natural Language Processing*, pages 8–14.

Yonatan Belinkov and James Glass. 2019. Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Cuong Xuan Chu, Niket Tandon, and Gerhard Weikum. 2017. Distilling task knowledge from how-to communities. In *Proceedings of the 26th International Conference on World Wide Web*, pages 805–814.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jeff Da and Jungo Kasai. 2019. Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations. *CoRR*, abs/1910.01157.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Aysu Ezen-Can. 2020. A comparison of LSTM and BERT for small corpus. *CoRR*, abs/2009.05451.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? In *CogSci*.

Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Cao Li, Qixun Zeng, Chengyue Sun, Yiyun Fei, Yu Zheng, Ying Li, Yi Liu, Peng Liu, Lin Ma, Le Weng, Xiaohang Hu, Xin Ma, Qian Qian, Rongfei Jia, Binqiang Zhao, and Hao Zhang. 2020. 3d-front: 3d furnished rooms with layouts and semantics. *arXiv preprint arXiv:2011.09127*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't. In *Proceedings of the NAACL-HLT SRW*, pages 47–54, San Diego, California, June 12-17, 2016. ACL.

Anthony G Greenwald, Debbie E McGhee, and Jordan LK Schwartz. 1998. Measuring individual differences in implicit cognition: the implicit association test. *Journal of personality and social psychology*, 74(6):1464.

Prakhar Gupta and Martin Jaggi. 2021. Obtaining better static word embeddings using contextual embedding models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5241–5253, Online. Association for Computational Linguistics.

William G Hayward and Michael J Tarr. 1995. Spatial language and spatial representation. *Cognition*, 55(1):39–84.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? (extended abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 5065–5069. International Joint Conferences on Artificial Intelligence Organization. Journal track.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Jad Kabbara and Jackie Chi Kit Cheung. 2021. Post-editing extractive summaries by definiteness prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3682–3692, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Stephen Kokoska and Daniel Zwillinger. 2000. *CRC standard probability and statistics tables and formulae*. Crc Press.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Barbara Landau and Ray Jackendoff. 1993. "what" and "where" in spatial language and spatial cognition. *Behavioral and Brain Sciences*, 16(2):217–238.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27:2177–2185.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2020. What does BERT with vision look at? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5265–5275, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6862–6868, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Prasanta Chandra Mahalanobis. 1936. On the generalized distance in statistics. National Institute of Science of India.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. 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 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

William Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *Transactions of the Association for Computational Linguistics*, 9:1047–1060.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. 2019. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8592–8600.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer.

Damien Sileo. 2021. Visual grounding strategies for text-only natural language processing. In *Proceedings of the Third Workshop on Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, pages 19–29, Kyiv, Ukraine. Association for Computational Linguistics.

Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. 2017. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*.

Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. 2007. Measuring and testing dependence by correlation of distances. *The annals of statistics*, 35(6):2769–2794.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *CoRR*, abs/1905.06316.

François Torregrossa, Vincent Claveau, Nihel Kooli, Guillaume Gravier, and Robin Allesiardo. 2020. On the correlation of word embedding evaluation metrics. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4789–4797.

Amos Tversky and Itamar Gati. 2004. Studies of similarity. In Eldar Shafir, editor, *Preference, Belief, and Similarity. Selected Writing of Amos Tversky*, pages 75–95. MIT Press.

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.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1671–1682, Berlin, Germany. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Li Zhang, Jun Li, and Chao Wang. 2017. Automatic synonym extraction using word2vec and spectral clustering. In *2017 36th Chinese Control Conference (CCC)*, pages 5629–5632. IEEE.

Wenxuan Zhou, Junyi Du, and Xiang Ren. 2019. Improving bert fine-tuning with embedding normalization. *arXiv preprint arXiv:1911.03918*.

A Appendix

A tabular breakdown of the datasets used can be seen in Table 4, 5 and 6. The exact models used are listed in Table 7. The heatmap visualizations for the other two datasets are in Figure 3 and 4.

A.1 Word Frequency

We also conducted an experiment to correlate the scores with their frequency. For this purpose, the corresponding objects of each target were selected. And then the distance correlation between the scores and the corresponding word frequency was calculated based on the average of the last 10 years of *Google Ngrams*. The results are shown in Table 9. The correlations are not particularly significant (mostly $p \geq 0.1$), but it is noticeable that especially the cosine score depends strongly on the word frequency. The classifiers are generally less sensitive.

bathroom		bedroom		kitchen		living room		office	
object	score	object	score	object	score	object	score	object	score
toilet	1.00	dresser	1.00	drying rack	1.00	coffee table	0.94	whiteboard	1.00
bathtub	1.00	night stand	1.00	kitchen island	1.00	ottoman	0.93	room divider	0.94
toothbrush holder	1.00	headboard	1.00	pot	1.00	fireplace	0.87	stapler	0.92
toothpaste	1.00	bed	0.97	frying pan	1.00	dvd player	0.69	cork board	0.92
shower curtain	1.00	alarm clock	0.97	spice rack	1.00	sofa	0.68	file	0.88
toothbrush	0.97	laundry basket	0.86	cutting board	1.00	decorative plate	0.61	keyboard	0.85
towel rod	0.96	hat	0.74	blender	1.00	tv stand	0.57	mouse	0.84
toilet paper	0.96	doll	0.70	knife	1.00	blanket	0.55	pen	0.83
squeeze tube	0.95	stuffed animal	0.60	stove	0.98	television	0.53	computer	0.82
faucet handle	0.82	pillow	0.56	dishwasher	0.97	remote control	0.50	column	0.81

Table 4: Statistics generated from ScanNet using NYU categories: *score* is the conditional probability $P(room \mid object)$ of the room given the object based on the frequencies observable in NYU.

bed		dishwasher		door		mortise lock		refrigerator		toilet	
object	score	object	score	object	score	object	score	object	score	object	score
pillow	1.00	drain hose	1.00	lock	1.00	ring	1.00	switch	1.00	valve seat shaft	1.00
bolster	1.00	overflow protection switch	1.00	cornice	1.00	keyway	1.00	refrigerator compartment	1.00	tank lid	1.00
mattress cover	1.00	tub	1.00	hanging stile	1.00	cotter pin	1.00	egg tray	1.00	conical washer	1.00
leg	1.00	pump	1.00	entablature	1.00	spring	1.00	shelf channel	1.00	lift chain	1.00
box spring	1.00	gasket	1.00	top rail	1.00	rotor	1.00	magnetic gasket	1.00	seat	1.00
headboard	1.00	water hose	1.00	middle panel	1.00	cylinder case	1.00	storage door	1.00	shutoff valve	1.00
mattress	1.00	heating element	1.00	bottom rail	1.00	key	1.00	freezer door	1.00	trip lever	1.00
pillow protector	1.00	rack	1.00	panel	1.00	faceplate	1.00	guard rail	1.00	ball-cock supply valve	1.00
elastic	1.00	cutlery basket	1.00	jamb	1.00	dead bolt	1.00	crisper	1.00	toilet bowl	1.00
footboard	1.00	wash tower	1.00	doorknob	1.00	cylinder	1.00	glass cover	1.00	flush handle	1.00
		motor	1.00	threshold	1.00	stator	1.00	butter compartment	1.00	wax seal	1.00
		detergent dispenser	1.00	weatherboard	1.00	strike plate	1.00	thermostat control	1.00	tank ball	1.00
		slide	1.00	lock rail	1.00			freezer compartment	1.00	float ball	1.00
		leveling foot	1.00	shutting stile	1.00			ice cube tray	1.00	filler tube	1.00
		insulating material	1.00	header	1.00			meat keeper	1.00	waste pipe	1.00
		spray arm	1.00					door stop	1.00	seat cover	1.00
		rinse-aid dispenser	1.00					shelf	1.00	cold-water supply line	1.00
								dairy compartment	1.00	overflow tube	1.00
								door shelf	1.00	trap	1.00
										refill tube	1.00

Table 5: A subset of part-whole relations extracted from *Online-Bildwörterbuch*. All parts have a value of *1.00* in our data set, because they only occur with this object.

eat		listen to		play		read		wash with		wear	
object	score	object	score	object	score	object	score	object	score	object	score
food	0.13	music	0.22	game	0.27	book	0.08	soap	0.29	clothing	0.07
diet	0.08	song	0.03	music	0.06	label	0.06	water	0.29	glove	0.06
meal	0.07	body	0.03	note	0.04	instruction	0.05	vinegar	0.04	shoe	0.05
breakfast	0.04	side	0.02	sport	0.03	review	0.04	solution	0.03	clothes	0.05
balanced diet	0.03	partner	0.02	chord	0.02	body language	0.02	detergent	0.03	shirt	0.02
fruit	0.03	child	0.02	song	0.02	rule	0.02	baking soda	0.03	makeup	0.02
vegetable	0.03	perspective	0.02	video game	0.02	example	0.02	cream	0.02	gear	0.02
plenty	0.03	response	0.02	card	0.02	complaint	0.01	shampoo	0.02	boot	0.02
protein	0.03	parent	0.02	role	0.02	law	0.01	towel	0.02	dress	0.02
snack	0.02	people	0.02	video	0.02	story	0.01	cold water	0.02	sock	0.02

Table 6: A subset of verb-object relations extracted from an updated version of HowToKB.

Model	Specification	Dimension	Parameters	Dataset Size (T ; S)	URL
word2vec	GoogleNews-vectors-negative300	300	-	100B ; -	https://code.google.com/archive/p/word2vec/
Glove	Common Crawl - glove.840B.300d	300	-	840B ; -	https://nlp.stanford.edu/projects/glove/
Levy	Dependency-Based Words	300	-	English Wiki (∼ 2B tokens)	https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings/
fastText	crawl-300d-2M-subword	300	-	600B ; -	https://fasttext.cc/docs/en/english-vectors.html
static-BERT	bert_12layer_sent	768	-	+1.28B ; -	https://zenodo.org/record/5055755
BERT-Base	bert-base-uncased	768	∼ 110M	3.3B ; 16GB	https://huggingface.co/bert-base-uncased
BERT-Large	bert-large-uncased	1024	∼ 336M	3.3B ; 16GB	https://huggingface.co/bert-large-uncased
RoBERTa	roberta-large	1024	∼ 336M	- ; 160GB	https://huggingface.co/roberta-large
ELECTRA	electra-large-generator	256	∼ 51M		https://huggingface.co/google/electra-large-generator
ALBERT	albert-xxlarge-v2	4096	∼ 223M	3.3B ; 16GB	https://huggingface.co/albert-xxlarge-v2
GPT2	gpt2-large	1280	∼ 774M	- ; 40GB	https://huggingface.co/gpt2-large
GPT-Neo	gpt-neo-2.7B	2560	∼ 2.7B	420B ; -	https://huggingface.co/EleutherAI/gpt-neo-2.7B
GPT-J	gpt-j-6B	4096	∼ 6B	- ; 825GB	https://huggingface.co/EleutherAI/gpt-j-6B

Table 7: Model overview. Mostly only the token quantity (T) or the dataset size (S) was given.

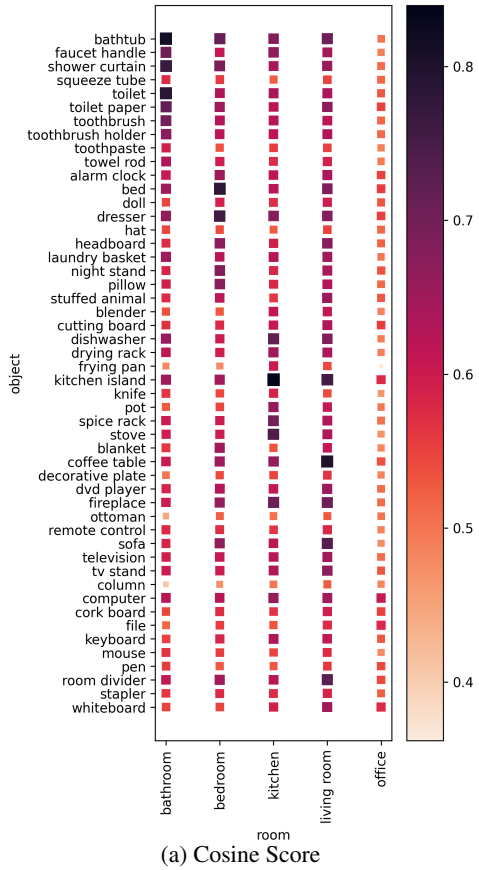
Task	Model	Data	Templates
Cosine Score & Classification	MLM & CLM	Room & Objects & Parts	This is a/an {x}. That is a/an {x}. There is a/an {x}. Here is a/an {x}. A/An {x} is here. A/An {x} is there.
		Verbs	I {x} something. I {x} anything. I {x}. You {x} something. You {x} anything. You {x}.
Increased Log Probability	MLM	Room & Object Object & Part Verb & Object	A/An {obj} is usually in the {room}. A/An {part} is usually part of a/an {obj}. I usually {verb} this {obj}.
	CLM	Room & Object	A/An {obj} is usually in the ... In the {room} is usually a/an ...
		Object & Part	A/An {part} is usually part of a ... In the {obj} is usually a/an ...
		Verb & Object	I usually {verb} this ...

Table 8: Templates for calculating scores regarding *Masked Language Models* (MLM) and *Causal Language Models* (CLM). For more details, see Sec. 4.

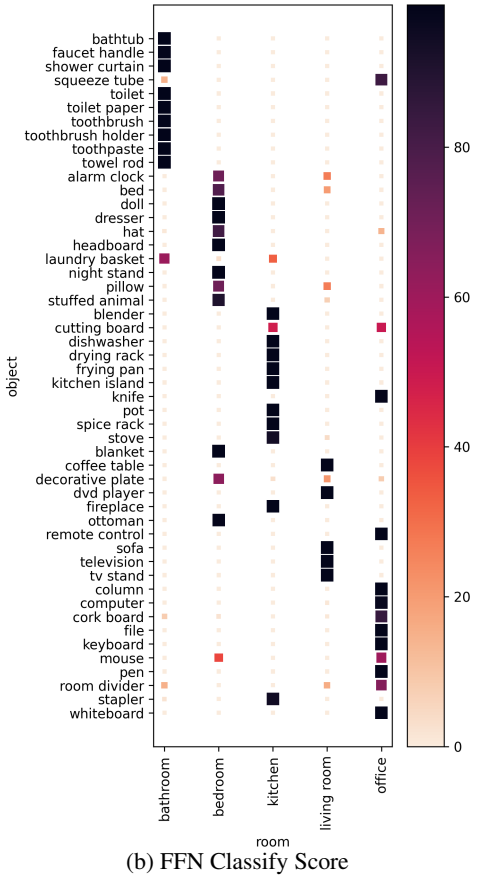
		Word2Vec						**GloVe**						**Levy**						**fastText**						**static-BERT**					
		cos	dist	kend	knn	svm	ffn	cos	dist	kend	knn	svm	ffn	cos	dist	kend	knn	svm	ffn	cos	dist	kend	knn	svm	ffn	cos	dist	kend	knn	svm	ffn
Room	bathroom	0.73	0.75	0.78	0.31	0.31	0.22	0.53	0.56	0.57	0.23	0.00	0.23	0.65	0.67	0.68	0.25	0.00	0.32	0.65	0.67	0.70	0.00	0.00	0.00	0.74	0.75	0.76	0.56	0.41	0.47
	bedroom	0.55	0.53	0.56	0.00	0.36	0.35	0.72	0.72	0.69	0.50	0.35	0.57	0.51	0.51	0.50	0.00	0.00	0.37	0.58	0.54	0.53	0.66	0.55	0.35	0.45	0.44	0.44	0.68	0.35	0.36
	kitchen	0.51	0.52	0.51	0.35	0.49	0.42	0.37	0.34	0.34	0.29	0.35	0.33	0.46	0.46	0.46	0.20	0.00	0.20	0.33	0.36	0.40	0.30	0.20	0.37	0.46	0.46	0.46	0.20	0.31	0.42
	living room	0.63	0.61	0.61	0.36	0.46	0.50	0.48	0.62	0.63	0.44	0.29	0.30	0.53	0.57	0.57	0.41	0.41	0.54	0.39	0.57	0.59	0.28	0.29	0.27	0.52	0.54	0.53	0.00	0.30	0.52
	office	0.65	0.65	0.58	0.38	0.26	0.37	0.52	0.57	0.56	0.38	0.43	0.47	0.66	0.66	0.65	0.27	0.50	0.50	0.45	0.43	0.41	0.58	0.35	0.30	0.37	0.40	0.40	0.57	0.17	0.42

		BERT-Base						**BERT-Large**						**RoBERTa**						**ElectraGen**						**Albert**					
		cos	m-s	m-t	knn	svm	ffn	cos	m-s	m-t	knn	svm	ffn	cos	m-s	m-t	knn	svm	ffn	cos	m-s	m-t	knn	svm	ffn	cos	m-s	m-t	knn	svm	ffn
Room	bathroom	0.40	0.35	0.30	0.23	0.23	0.23	0.52	0.34	0.50	0.23	0.23	0.24	0.61	0.47	0.42	0.43	0.23	0.35	0.58	0.66	0.39	0.35	0.35	0.40	0.69	0.56	0.36	0.34	0.36	0.39
	bedroom	0.61	0.47	0.41	0.45	0.28	0.37	0.63	0.36	0.37	0.42	0.19	0.36	0.67	0.56	0.33	0.42	0.28	0.41	0.41	0.53	0.62	0.68	0.55	0.50	0.54	0.58	0.36	0.18	0.31	0.47
	kitchen	0.34	0.60	0.35	0.30	0.23	0.43	0.38	0.73	0.45	0.75	0.75	0.62	0.65	0.38	0.49	0.46	0.34	0.24	0.48	0.37	0.37	0.25	0.44	0.43	0.43	0.75	0.38	0.35	0.22	0.45
	living room	0.52	0.57	0.56	0.25	0.20	0.32	0.47	0.47	0.54	0.36	0.36	0.39	0.43	0.51	0.45	0.51	0.27	0.27	0.41	0.36	0.45	0.44	0.49	0.47	0.38	0.54	0.47	0.19	0.30	0.34
	office	0.68	0.56	0.64	0.35	0.44	0.56	0.64	0.75	0.67	0.26	0.58	0.45	0.48	0.49	0.53	0.27	0.28	0.47	0.50	0.70	0.55	0.59	0.55	0.25	0.72	0.54	0.61	0.58	0.37	0.36

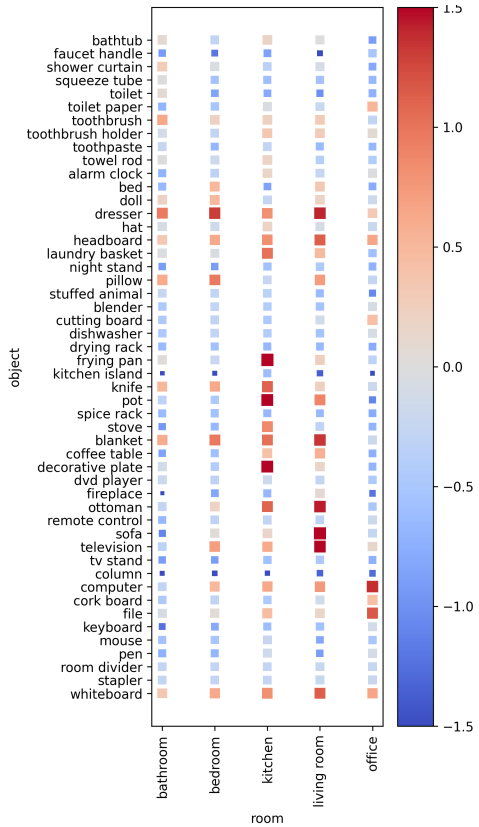
Table 9: Distance Correlation calculated on the word frequencies of Google Ngram. (A gray cell indicates significant at $p < 0.1$)



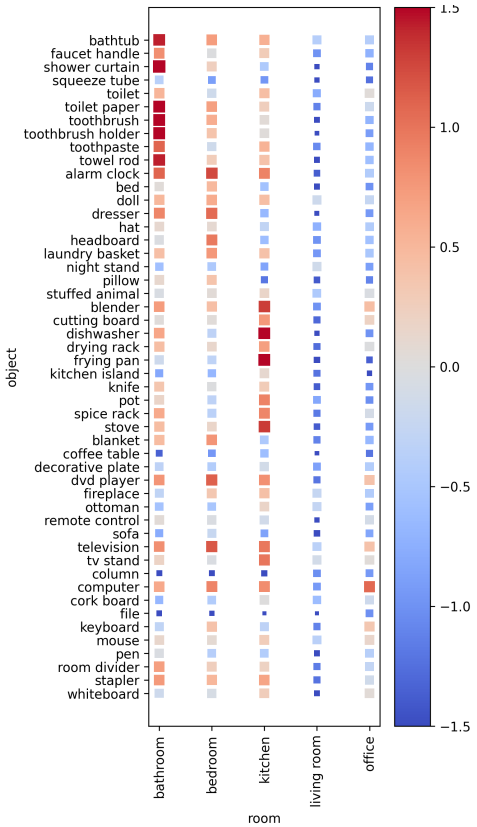
(a) Cosine Score



(b) FFN Classify Score

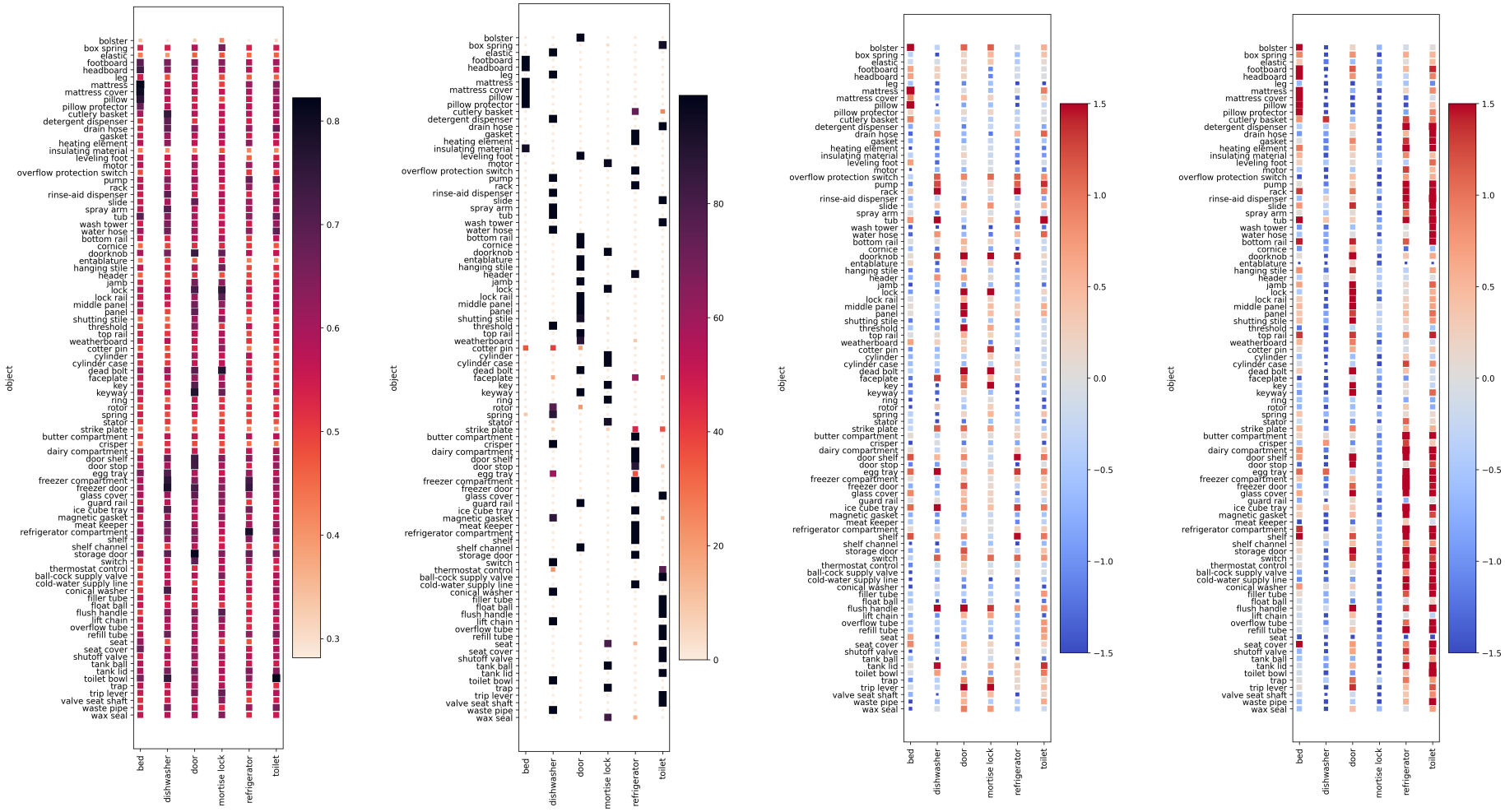


(c) Mask Object Score



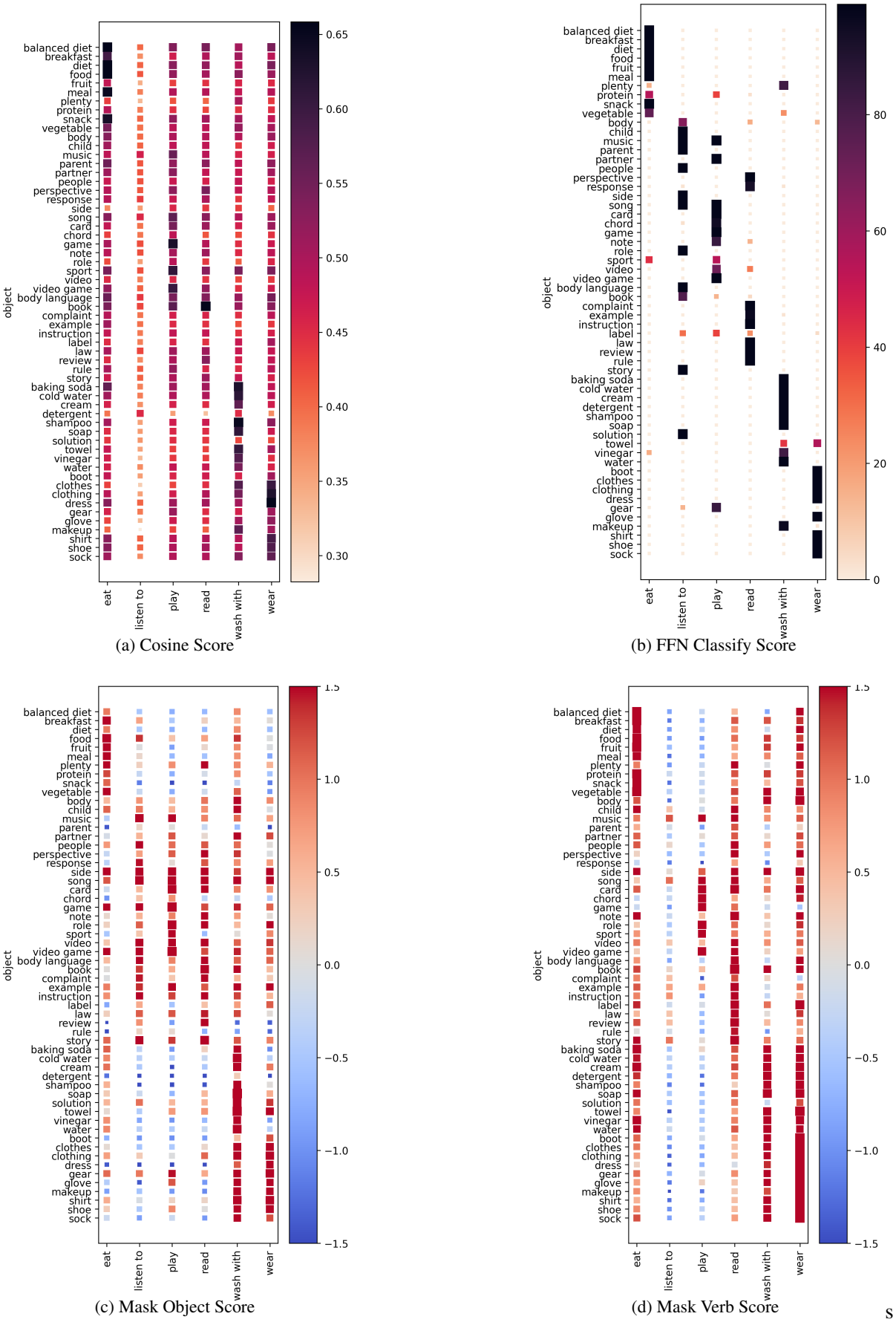
(d) Mask Room Score

Figure 2: Heatmap of source-object associations based on BERT-Large and the room dataset. The objects (sources) on the y-axis are grouped by the room in which they are most likely to be located according to the *NYU Depth V2 Dataset*.



(a) Cosine Score (b) FFN Classify Score (c) Mask Part Score (d) Mask Object Score

Figure 3: Association heatmap of BERT-Large on the part dataset. The parts (sources) on the y-axis are grouped by the room in which they are most likely to be located according to the *Online-Bildwörterbuch Dataset*.



(a) Cosine Score

(b) FFN Classify Score

(c) Mask Object Score

(d) Mask Verb Score

Figure 4: Association heatmap of BERT-Large on the verb dataset. The objects (sources) on the y-axis are grouped by the room in which they are most likely to be located according to the *HowToKB Dataset*.