Probing Large Language Models for Scientific Synonyms

Freya Thießen[1,*], Jennifer D'Souza[1] and Markus Stocker[1]

[1] TIB–Leibniz Information Centre for Science and Technology, Welfengarten 1B 30167 Hannover

Abstract

Purpose: Automatically identifying synonyms is an important but challenging aspect of entity normalization in knowledge graphs. Entity normalization is crucial in ensuring that information in knowledge graphs is well connected and therefore efficiently reusable. We aim to investigate the potential of pre-trained large language models (LLMs) for this task.

Methodology: We use k-Means clustering to compare latent concepts learned by LLMs with human-defined scientific synonymy concept clusters sourced from ORKG, CS-KG, SemEval 2017, and SciERC data. We investigate the models BERT, RoBERTa, BART, and OpenAI GPT3 (text-embedding-ada-002 variant) and evaluate clustering results by model layer.

Findings: F_1 scores average around 0.7 to 0.75 depending on the dataset and layer. The best results are reached using OpenAI GPT3 (max F_1=0.914). We further notice no advantage of models trained on scientific data.

Value: Our results suggest information learned by transformer models aligns with human-defined scientific synonyms. This shows the potential of information encoded in pre-trained LLMs to be leveraged for synonymy detection.

Keywords

entity normalization, synonymy, LLMs, knowledge graphs

1. Introduction

Knowledge graphs (KGs) can help to represent, organize, make accessible, and more efficiently reusable the large and ever-increasing amount of scientific findings. However, in order to do so, it is vital that the entities they contain are richly described and well connected.

One possible way to build rich KGs is through leveraging existing human knowledge by employing crowd-sourcing. The Open Research Knowledge Graph (ORKG, https://orkg.org/), for example, is a community-curated KG for scientific information extracted from journal publications, conference proceedings, and other academic literature [1, 2]. Its web-based frontend interface supports researchers in creating semantic descriptions of their scientific findings. Hence, users collaboratively populate and maintain the knowledge graph.

An important challenge in crowd-sourced KGs is the appearance of non-connected *name variations* of entities in the graph. Name variations are different wordings, spelling differences

SEMANTICS 2023 EU: 19th International Conference on Semantic Systems, September 20-22, 2023, Leipzig, Germany

[*] Corresponding author.

freya.thiessen@tib.eu (F. Thießen)

0009-0000-2513-6543 (F. Thießen); 0000-0002-6616-9509 (J. D'Souza); 0000-0001-5492-3212 (M. Stocker)

or synonyms of the same underlying entity. Fakhraei et al. [3] differentiate between *syntactic* and *semantic variations.* Syntactic variations cover spelling differences, changes in word order or capitalization and similar (e.g., 'FOXP2' and 'FOX-P2') while semantic variations are significant changes in naming not linkable to the same entity by considering syntactical changes that a human can, however, still identify as describing the same entity (e.g., 'P70 S6KA' and '52 kDa ribosomal protein S6 kinase'). Name variations — especially semantic variations — pose a risk to richly described, dense KGs, because they can lead to missing links between information about the same underlying entity. A crucial task in knowledge graph curation is therefore the normalization of entities and linking between semantic variations.

The linguistic property we are most interested in here is synonymy. Synonyms are different terms, which express the same fact and cover both the same meaning as well as extension. They can be differentiated from hyper- and hyponyms, which form a generalization or specialization, respectively.

Our aim is to measure to what degree the semantic information encoded in large language models (LLMs) aligns with the linguistic property of synonyms. Specifically, we are interested in whether concept clusters can be automatically (re-)constructed from LLMs. A *concept* or *concept cluster* is understood here as a set of synonymous scientific terms. In this way LLMs can aid normalization of knowledge graphs by identifying missing relations between semantic variations of entities. More generally, our work provides foundational empirical validation that can support future work in automatic ontology discovery from LLMs through the discovery of specific relations (in this case synonymy).

Large language models are typically neural networks with large sets of parameters, trained on a language modeling task – assigning probabilities to the next token in a sentence [4]. Many models are able to represent some form of general language capability, applicable to a wide range of tasks without extensive re-training. LLMs have been found to replicate representations of linguistic structures ranging from syntax trees [5] to tense [6]. Effectively, LLMs have become a standard for a wide range of practical NLP applications such as machine translation or entity linking [7, 8].

We aim to analyze whether the word representation built by LLMs encode linguistic information relevant for detecting synonyms, specifically for scientific entities. Additionally, we investigate which model and network layers show the best prospect for the synonymy detection task, in order to guide future work on developing a knowledge graph normalization use case for this task. For this purpose we consider SciBERT, RoBERTa, BART, and OpenAI GPT3 (text-embedding-ada-2 variant) as models. In detail, we aim to answer the following research questions (**RQ**s):

RQ1. How well do word embeddings encoded by LLMs capture the linguistic property of synonymy?

RQ2. Which LLMs best encode synonymy information?

RQ3. At which layers in the LLM network are the most effective embeddings for synonymy found?

2. Related Work

Our approach is situated within the task of probing contextualized word representations for semantic linguistic properties, namely synonymy relations, by means of a cluster analysis of the embeddings. In this section we discuss related work. We conclude that correctly identifying synonyms is one of the most challenging aspects of the normalization task. Previous experiments probing LLMs show that semantic representations exist and are similar to linguistic categories in mid- to higher layers of the models. Additionally, cluster analysis can be used to uncover latent concepts encoded in the representations. This provides an avenue to investigate whether latent concepts in the representations produced by LLMs encode synonymy.

2.1. Entity Normalization

Fakhraei et al. [3] define entity normalization as the task of linking a string to a canonical reference set. They use a deep Siamese neural network to employ similarity learning and embeddings for entity normalization. The authors empirically validate their approach on biomedical data, but argue that the approach is generalizable beyond this domain.

Another example of entity normalization specific to the biomedical domain can be found in Sung et al. [9]. The authors base their work on BioBERT, which is a BERT-based model trained on biomedical data [10]. Sung et al. [9] use a synonymy marginalization technique which is supposed to maximize the probability of synonyms in the entity candidate selection via marginal probabilities. They identify synonyms with different surface forms as well as entities with similar surface forms but different meanings as the most challenging aspect of the normalization task. They argue that this calls for latent representations capturing semantic meaning for entity normalization.

2.2. Probing of LLMs

Previous works have investigated contextual word representations for a number of linguistic phenomena, ranging from syntactic to semantic. We explicitly limit our description of existing literature here to investigations of neural models trained on language modeling tasks, such as large transformer models like BERT, although similar probing research exists for other neural network models as well [e.g., 11, 12].

Hewitt and Manning [5] have investigated whether ELMo and BERT implicitly encode syntax trees through their vector geometry by employing a so-called structural probe. This probe identifies linear transformations of the word embedding space, which encode distance and depth of a parse tree. They have found evidence that this linguistic structure is present and recoverable from word representations in vector space.

Liu et al. [13] employ a range of probing tasks to contextualized word representations ranging from token labeling (e.g. part-of-speech-tagging POS) or segmentation (e.g. named entity recognition) to pairwise word relations of both syntactic as well semantic dependency between words. They note that high performance is possible on a broad range of task with the features generated from pre-trained LLMs and conclude that this means they encode useful and transferable features of language.

Both Tenney et al. [14] and Jawahar et al. [6] probe BERT with a set of probing tasks applied on the basis of extracted contextualized vectors from all network layers. Tenney et al. [14] test on POS, coreference, and relation classification, among others, while Jawahar et al. [6] test for a range of features ranging from surface level (e.g. sentence length) and syntactic (e.g. syntactic tree depth) to semantic (e.g. tense). Both studies find evidence that BERT encodes this information with processing similar to the classic NLP pipeline [14], meaning ordered from local syntactic in lower layers to more semantic and longer-ranged dependencies in higher layers.

Pimentel et al. [15] argue that the highest performing probe should be chosen even if it is more complex, because it can give a better estimate of the full information available through the model. This is in contrast to e.g. Liu et al. [13], who base their analysis on training simple probe models arguing that if a simple model can make predictions one can reasonably assume that this performance is based on the underlying information present in the pre-trained LLM rather than through any additional complexity available through the probe. Pimentel et al. [15] probe BERT contextualized embeddings for POS and dependency labeling in eleven languages with a focus on the amount of information extractable compared to non-contextualized baselines. They find that while BERT does in fact encode syntactic information needed for POS and dependency labeling, the advantages compared to baseline are at most 12% more information, and differ across languages.

2.3. Clustering Analysis

Sajjad et al. [16] analyze transformer models in terms of human-defined concepts through clustering. They attempt to align clusters of contextual embeddings with human-defined concepts (e.g. POS tags, synonym sets) to explain the concept representation in the model. Using this approach, Sajjad et al. [16] find that clusters in lower layers more strongly align with lexical concepts, while morphological and syntactical concepts are better represented in middle to higher layers. The alignment with linguistic ontologies (i.e. WordNet synonymy sets) is stronger in lower to middle layers and declines for higher, more contextualized layers.

Dalvi et al. [17] define groupings of word representations based on syntactic and semantic relations as *latent concepts*, 'the information the model learns about language' [17, p.1]. They investigate these latent concepts in pre-trained BERT and analyze them in comparison to traditional linguistic concepts and across model layers. Based on their results, they conclude that the model learns both novel concepts not directly associated with pre-defined linguistic categories as well as concepts based on multiple semantic, syntactic, and morphological concepts. Additionally, the authors observe lower layers are dominated by shallow lexical concepts whereas higher layers represent semantic relations.

3. Methods

We created and analyzed four different sets of data. All data taken together consist of 25 214 entities organized in 3746 concept clusters. Selected summary statistics about the data are shown in table 1.[1]

Table 1
Selected summary statistics for the four datasets evaluated in this work.

parameters	ORKG	CS-KG	SemEval 2017	SciERC
entities	247	23611	687	669
clusters	117	3000	336	293
avg. ent./ cluster	2.11	7.87	2.04	2.28
domains	various	CS	CS, material sci., physics	AI

3.1. Data Sources

In our experiments, we leveraged existing relevant datasets that provided gold-standard synonymy annotations as far as possible. This was advantageous in two respects: 1) reusing existing datasets significantly alleviates the high cost barrier, especially in terms of effort, to obtain new gold-standard annotations for the task we address, and 2) existing datasets can be benchmarked as standards. Our goal for data collection was to reach a sufficiently large domain-diverse set of scientific synonyms.

ORKG As introduced above, the ORKG is a community-curated knowledge graph for scientific information extracted from journal publications, conference proceedings, and other academic literature [1, 2]. The ORKG is open to research from all domains ranging from life sciences to computer science or social sciences; hence, extracted data are highly diverse. To construct the ORKG dataset used in our study, we selected all entity pairs connected via a 'same as' relation.[2]

CS-KG The second dataset utilizes the Computer Science Knowledge Graph (CS-KG) [18], which is a large-scale automatically generated knowledge graph based on scientific publications in computer science. CS-KG provides sets of alternative names for all entities contained in it. These are taken to be synonyms and the sets of alternative names used as concepts without further processing.

SemEval 2017 SemEval 2017 Task 10 was a task to extract keyphrases and relations from scientific text [19]. The task data comes from journal articles in the domains of computer science, materials science, and physics. Selected paragraphs were annotated with hypernym and synonym relations. We use synonym pairs from both tagged train and test data for our dataset.

[1] Data are available at https://zenodo.org/record/7971572.
[2] https://orkg.org/property/SAME_AS The selection reflects the state of the ORKG data as of November 2022.

SciERC The SciERC dataset is an extension of SemEval 2017 Task 10 and SemEval 2018 Task 7 [20]. It was created by annotating abstracts from AI research and consists of annotated data for entity, relation, and co-reference classification. Due to the fact that a synonymy relation was not part of the annotations for SciERC we sourced our data from annotated co-references in train, test, and development data.

3.2. Data Cleaning

The collected data consisted of synonym pairs for ORKG and SemEval 2017 but co-reference pairs for SciERC. In order to create synonym concepts we performed data cleaning and then combined synonym pairs into larger concepts.

We lower-cased all data extracted from ORKG, SemEval 2017, and SciERC. Further, we removed synonymy pairs where both entities were identical strings as well as duplications. The ORKG data included entities consisting of hyperlinks, which were removed. All remaining co-reference pairs from SciERC were considered potential synonyms. Coreference translates to synonymy when the relation holds between noun phrases or nouns, but in other cases, for instance coreference to articles, the relation does not translate to synonymy. Thus the given dataset had to be appropriately filtered to suit our synonym identification task. From the pool of co-reference pairs we removed all pairs where either one entity was a generic reference (e.g. *this*) or one entity was a generalization of the broader second entity (e.g. *normalization method* - *method*). Additionally, we performed a manual curation of ORKG and SciERC data to remove incorrect of irrelevant pairs.

Finally, we combined the synonym pairs into larger concept clusters wherever possible. This was done by identifying entity overlaps between pairs where one entity would occur in several synonym pairs (e.g. *data set* - *dataset*; *data* - *data set*). Each resulting concept cluster was assigned a randomly chosen entity from the cluster as preferred name for the entire cluster and a whole number as ID.

3.3. Models

Four LLMs were selected for this study: SciBERT, RoBERTa, BART, and OpenAI GPT3 (text-embedding-ada-002 variant) model. Both SciBERT and RoBERTa are transformer models based on the BERT architecture and trained on a masked language task [21, 22]. We probe the 12-layer versions `roberta-base` and `scibert_scivocab_uncased`. BART is a transformer encoder-decoder (seq2seq) model; the authors describe the architecture as a generalization of BERT and GPT-based systems [23]. The model is trained on both corrupting text with noise and reconstructing the original text. We use the 6-layer version `bart-base`. The OpenAI model we used for investigation — i.e., 'text-embedding-ada-002' — is a GPT3 based model specifically adapted for embedding generation[3]. In terms of training materials, SciBERT differs from the other models as it is trained on scientific publications while RoBERTa and BART are trained on a range of books, news material, and web content. There is unfortunately no detailed information publicly available about the training material used for OpenAI's model.

[3] https://platform.openai.com/docs/models/overview

3.4. Clustering and evaluation

Based on the datasets described above, word embeddings are calculated for all scientific entities. Each entity is treated as a separate sentence and encoded by the model independently of each other. For SciBERT, RoBERTa, and BART token embeddings are extracted from the models by network layer and aggregated to a single word embedding per layer by applying a mean function across token embeddings. This is done using Python libraries transformers [24, version 4.23.1] and numpy [25, version 1.23.2]. For OpenAI's GPT3 model the API provided by OpenAI is used to access final layer word embeddings for the scientific entities in the datasets.

The analysis is performed on the basis of the resulting word embeddings. A kMeans algorithm is applied with the number of concept clusters in each dataset given as parameter k. This means that, for example, the algorithm is run with $k = 3000$ for CS-KG data and $k = 117$ for the ORKG dataset. We used the kMeans implementation from Python package scikit-learn [26, version 1.1.2]. We chose kMeans as a method of analysis, because it is a low resource compute algorithm offering sound results for any starter investigative work on the theme around clustering. Subsequently, an F_1 score is adapted for the task and used to evaluate the clustering results. Here, precision is defined as

$$precision = max(\frac{tp_{cluster}}{n_{cluster}}) \tag{1}$$

where $tp_{cluster}$ is the number of entities from the same concept appearing in the same cluster and $n_{cluster}$ the total number of entities in that cluster. Recall, on the other hand, is defined here as follows:

$$recall = max(\frac{tp_{cluster}}{n_{concept}}) \tag{2}$$

where $tp_{cluster}$ is defined as above and $n_{concept}$ is the total number of entities in the relevant concept. In both cases, the highest possible value reached across all concepts and clusters is taken to be the recall or precision value for the clustering.

As an example, assume concept A with 3 entities (*data*, *dataset*, *data set*), concept B with 2 entities (F_1, F_1 *score*) and 2 clusters x (*data*, F_1) and y (*dataset*, *data set*, F_1 *score*). It follows that, for concept A $precision = max(\frac{1}{2}, \frac{2}{3}) = 0.75$ and $recall = max(\frac{1}{3}, \frac{2}{3}) = 0.75$. For concept B, $precision = max(\frac{1}{2}, \frac{1}{3}) = 0.5$ and $recall = max(\frac{1}{2}, \frac{1}{2}) = 0.5$. F_1 is then calculated as the harmonic mean of precision and recall, in this case $F_1 = 0.5$.

4. Results

Tables 2 to 5 show the F1-scores achieved by kMeans clustering the embeddings extracted from SciBERT, RoBERTa, BART, and OpenAI's GPT3 models, respectively. The data is organized by layer wherever embeddings are extracted from several hidden layers of the model.

The best F_1 scores for all four datasets are achieved using OpenAI GPT3 ($\bar{F_1} = 0.756$). Following are SciBERT ($\bar{F_1} = 0.713$), RoBERTa ($\bar{F_1} = 0.691$), and BART ($\bar{F_1} = 0.683$). Looking beyond the aggregated scores across all datasets, the order of highest scoring model changes slightly depending on the data: For CS-KG, SemEval 2017, and SciERC data the next best

Table 2
F_1 scores of kMeans clustering by SciBERT layer. Rounded to three decimals, highest score across layers for each dataset marked in bold.

SciBERT layer	*dataset*			
	ORKG	CS-KG	SemEval 2017	SciERC
layer 1	0.601	0.897	0.573	0.713
layer 2	0.589	**0.902**	0.570	**0.731**
layer 3	0.616	0.895	**0.576**	0.702
layer 4	0.577	0.884	0.568	0.701
layer 5	0.599	0.875	0.561	0.695
layer 6	0.618	0.859	0.559	0.686
layer 7	0.623	0.813	0.556	0.682
layer 8	0.621	0.788	0.554	0.673
layer 9	0.628	0.786	0.561	0.674
layer 10	**0.644**	0.766	0.553	0.67
layer 11	0.623	0.791	0.561	0.681
layer 12	0.613	0.782	0.572	0.681

Table 3
F_1 scores of kMeans clustering by RoBERTa layer. Rounded to three decimals, highest score across layers for each dataset marked in bold.

RoBERTa layer	*dataset*			
	ORKG	CS-KG	SemEval 2017	SciERC
layer 1	0.598	0.899	**0.573**	0.657
layer 2	0.582	**0.902**	0.57	0.681
layer 3	0.584	0.897	0.573	0.677
layer 4	0.593	0.88	0.556	**0.684**
layer 5	0.593	0.868	0.552	0.673
layer 6	0.589	0.854	0.557	0.667
layer 7	0.587	0.842	0.554	0.659
layer 8	0.578	0.813	0.553	0.643
layer 9	0.571	0.807	0.562	0.644
layer 10	0.579	0.799	0.558	0.639
layer 11	0.582	0.807	0.556	0.639
layer 12	**0.604**	0.793	0.561	0.634

performance is achieved using SciBERT, followed by RoBERTa and then BART models. The order of best perfomance for the ORKG data is OpenAI GPT3, SciBERT, BART, and RoBERTa.

Generally, the clustering approach works best based on the CS-KG dataset across all models (best F_1: 0.914), followed by SciERC (0.816), ORKG (0.698), and SemEval 2017 data (0.597). This ranking is independent of the model. When comparing the results from all four models and datasets, it can be noted that very similar performances are achieved across models. The biggest

Table 4
F_1 scores of kMeans clustering by BART layer. Rounded to three decimals, highest score across layers for each dataset marked in bold.

BART layer	*dataset*			
	ORKG	CS-KG	SemEval 2017	SciERC
layer 1	**0.618**	0.789	**0.573**	**0.668**
layer 2	0.604	0.784	0.572	0.651
layer 3	0.611	0.765	0.568	0.65
layer 4	0.597	0.752	0.559	0.642
layer 5	0.605	0.803	0.565	0.644
layer 6	0.597	**0.874**	0.562	0.663

Table 5
F_1 scores of kMeans clustering for OpenAI GPT3. Results rounded to three decimals.

dataset			
ORKG	CS-KG	SemEval 2017	SciERC
0.698	0.914	0.597	0.816

difference in F_1 score between models for the ORKG data is between RoBERTa and OpenAI GPT3, which is a score difference of 0.094 (only focusing on the layer with best performance for the model). Similarly, the biggest difference in best F_1 score across models is 0.04 for CS-KG, and 0.024 for SemEval 2017 data. For results based on SciERC data a slightly larger performance difference between models can be observed. Between the score achieved on GPT3 (0.816) and the best score achieved on the lowest performing model BART (0.668) lies a difference of 0.148. Generally, the performance changes depend much more strongly on the underlying data than the utilized model. These results are visualized in 1.

In an attempt to better understand our results, we calculated Pearsons's r between the best F_1 score achieved by each dataset and the average number of entities per cluster. We found that the score is highly correlated with the avg. number of entities per cluster for all models we tested: $r = 0.908$, $r = 0.96$, and $r = 0.966$, $r = 0.783$ for SciBERT, RoBERTa, BART, and OpenAI GPT3, respectively.

For all but OpenAI GPT3 we were able to extract embeddings from all network layers and compare clustering results achieved on them. In general, the best scores were achieved on embeddings extracted from one of the first three to four network layers, but there are a few noticeable exceptions to this pattern. Firstly, results based on the ORKG data were best on later layers for SciBERT and RoBERTa (10 and 12), but not for BART, where performance was best on the first layer. In contrast, best performance for CS-KG data was achieved on layer two for SciBERT and RoBERTa, but on the last layer six for BART. Figure 2 shows an illustration of these opposing patterns. Additionally, it can be noted that there are larger performance differences by layer for the CS-KG data overall, compared to the other datasets.

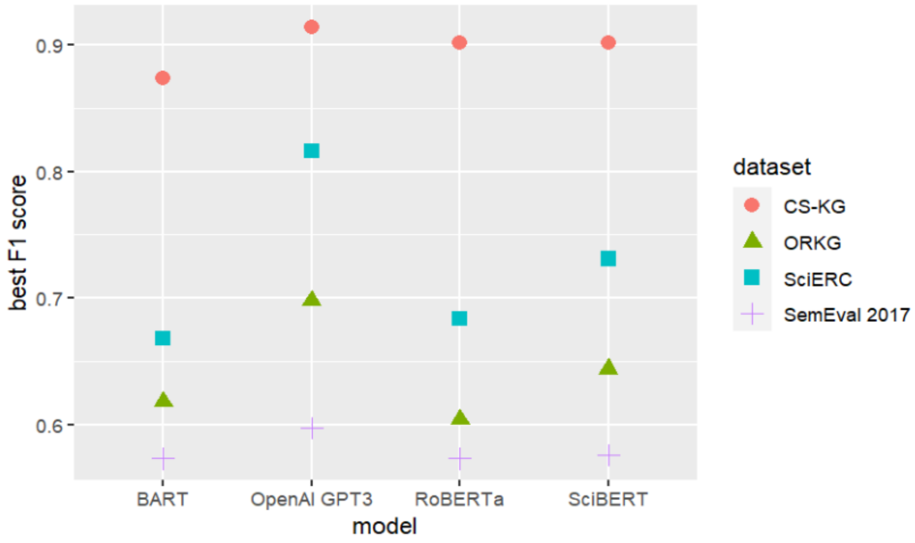

Figure 1: Best F_1 score performances organized by model and dataset

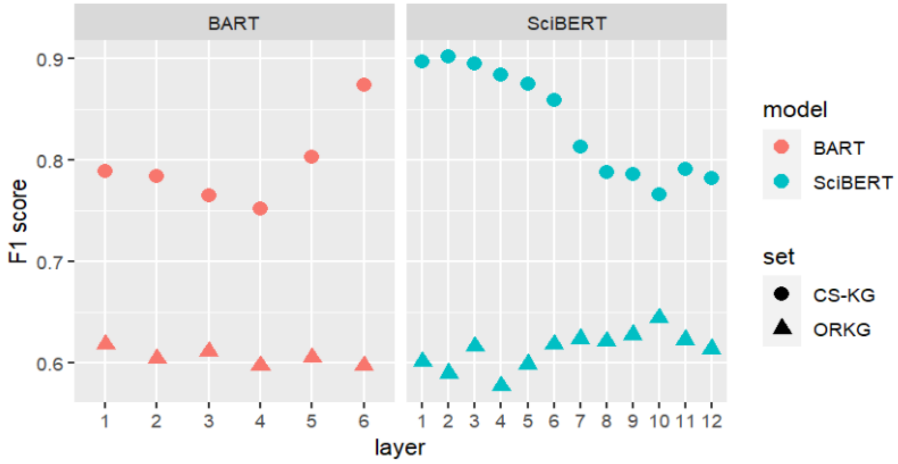

Figure 2: F_1 score results for ORKG and CS-KG data using models BART and SciBERT, illustrated as scatter plot.

5. Discussion

We have probed word embeddings encoded by LLMs for synonymy information by comparing clusters of embeddings, identified by kMeans, to pre-defined concept clusters of synonymous scientific entities from four different datasets. Answering our RQ1, we report F_1 performances averaging around 0.7 to 0.75 across datasets and tested models. Our results suggest that synonymy information is present and salient in the embeddings and can be leveraged through clustering to correctly identify synonym sets. We conclude from these results that LLMs encode information inline with the linguistic property of synonymy and that it is possible to leverage this information to detect sets of synonymous scientific entities from their embedding representation.

We further identify that the quality of the detection seems to depend much more strongly on the underlying data than on the model employed for the task. In our experiments concerning RQ2, OpenAI GPT3 gave the best performance, followed by SciBERT and RoBERTa with only a slight drop in performance. Of all tested models, only SciBERT is explicitly trained on scientific publications. While we expected this to be an advantage for encoding synonymous scientific entities, we observe only a slightly better clustering performance based on SciBERT-encoded embeddings compared to RoBERTa and BART and the best performance overall from OpenAI's GPT3 model. Our analysis suggests that there is at most a small advantage of training on scientific data compared to non-scientific data for the task of recovering synonymy structure of scientific entities. With the methodology employed here, we cannot fully explore why this is the case, but it might be that sufficiently large scale models encounter enough relevant information during training in order to not need specialization for scientific language for the synonymy detection task. Due to better availability of large-scale models trained on non-scientific text data, our findings could inform future research to utilize these models as an avenue for the synonymy detection task for scientific text data.

Performance differences for the different datasets seem to depend at least in part on the size of the concepts and potentially on other unidentified factors. One explanation for the effect of concept size could be that larger concepts, not limited to two or three entities, might provide more information on the target concept, guiding the clustering process. An additional potential factor in the performance differences might be related to morphological similarity in the concepts. The CS-KG dataset, based on which best clustering performance is achieved, contains some concepts structured as follows: A scientific entity (e.g. activation function) is contained in the concept set with a number of variations that share the same initial term but are supplemented with additional more generic terms (e.g. activation function process, activation function processing). This means that in these cases concepts have not only a strong semantic similarity but also a strong morphological similarity. Possibly, the high clustering performance needs to be in part attributed to this additional available information.

Concerning RQ3 we could not reach conclusive evidence as to which network layers are best suited for querying synonymy information. We observe that the optimal layer depends both on the model as well as the dataset. With the exception of the ORKG data embedded by SciBERT or RoBERTa and the CS-KG data embedded with BART we reach optimal performance in the lower to mid-ranger layers (≤ 4). This is congruent with findings by Sajjad et al. [16], who find a stronger alignment between embedding clusters and WordNet synonymy sets in lower to middle layers than in higher layers, and Jawahar et al. [6], who identify encoding of semantic information in mid-range layers. The different layer optimum for the ORKG dataset (layer 10 for SciBERT, 12 for RoBERTa) could speak to an underlying structural difference in the data compared to SemEval 2017 and SciERC. Further investigation is needed to identify the reasons for theses differences.

5.1. Limitations

Only some of the data used in this study was specifically annotated with regard to the synonymy relation. Specifically, from SciERC we base our further selection on co-reference data. We performed both automatic as well as manual data cleaning to correctly identify synonym pairs contained in the co-reference data, but manual cleaning was done in only one round by a single curator. It is likely that there are remaining instances where data overlaps with hyper- or hyponym relations and further data cleaning efforts would increase the data quality with regard to only containing strictly defined synonymy relations.

While we have chosen to investigate models that can create contextualized word embeddings, for this study we had to limit the analysis to pseudo-static representations nonetheless. This was done because some of the datasets used for our experiments contained only the scientific entities without any additional sentence context in which they appeared (namely ORKG and CS-KG data). To keep conditions comparable across all tested data we opted to embed all terms as free-standing entities outside of their sentence context.

5.2. Future Work

We have provided some evidence that semantic information inline with synonymy is implicit in LLMs and can be leveraged to recover synonymy structure from word embeddings. Further research could build on the findings presented here in two directions. One possibility is to extend the research to other types of relations (e.g. antonyms, hyponyms etc.). The other possibility is to use these findings to explore practical applications of the approach presented here for the task of knowledge graph normalization. Further work could e.g. fine-tune LLMs for use in clustering of existing KG entities. The analysis presented here, utilizing kMeans as a clustering algorithm on embeddings, likely does not capture the full extent of available synonymy information present in the investigated LLMs. As a comparatively simple approach, it should rather be considered a lower boundary for the task, which can be improved with additional probing experiments and additional LLM task fine-tuning.

Acknowledgments

This work was co-funded by the European Research Council for the project ScienceGRAPH (Grant agreement ID: 819536) and TIB–Leibniz Information Centre for Science and Technology.

References

[1] S. Auer, A. Oelen, M. Haris, M. Stocker, J. D'Souza, K. E. Farfar, L. Vogt, M. Prinz, V. Wiens, M. Y. Jaradeh, Improving access to scientific literature with knowledge graphs 44 (2020) 516–529. URL: https://www.degruyter.com/document/doi/10.1515/bfp-2020-2042/html. doi:10.1515/bfp-2020-2042.

[2] M. Stocker, A. Oelen, M. Y. Jaradeh, M. Haris, O. A. Oghli, G. Heidari, H. Hussein, A.-L. Lorenz, S. Kabenamualu, K. E. Farfar, M. Prinz, O. Karras, J. D'Souza, L. Vogt, S. Auer, Fair

scientific information with the open research knowledge graph, FAIR Connect 1 (2023) 19–21. URL: http://dx.doi.org/10.3233/fc-221513. doi:10.3233/fc-221513.

[3] S. Fakhraei, J. Mathew, J. L. Ambite, NSEEN: Neural semantic embedding for entity normalization, in: U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, C. Robardet (Eds.), Machine Learning and Knowledge Discovery in Databases, volume 11907, Springer International Publishing, 2020, pp. 665–680. URL: http://link.springer.com/10.1007/978-3-030-46147-8_40. doi:10.1007/978-3-030-46147-8_40, series Title: Lecture Notes in Computer Science.

[4] D. Jurafsky, J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3 - draft ed., 2023.

[5] J. Hewitt, C. D. Manning, A structural probe for finding syntax in word representations, in: Proceedings of the 2019 Conference of the North, Association for Computational Linguistics, 2019, pp. 4129–4138. URL: http://aclweb.org/anthology/N19-1419. doi:10.18653/v1/N19-1419.

[6] G. Jawahar, B. Sagot, D. Seddah, What does BERT learn about the structure of language?, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019, pp. 3651–3657. URL: https://www.aclweb.org/anthology/P19-1356. doi:10.18653/v1/P19-1356.

[7] M.-T. Luong, M. Kayser, C. D. Manning, Deep neural language models for machine translation, in: Proceedings of the Nineteenth Conference on Computational Natural Language Learning, 2015, pp. 305–309.

[8] Ö. Sevgili, A. Shelmanov, M. Arkhipov, A. Panchenko, C. Biemann, Neural entity linking: A survey of models based on deep learning, Semantic Web (2022) 1–44.

[9] M. Sung, H. Jeon, J. Lee, J. Kang, Biomedical entity representations with synonym marginalization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 3641–3650. URL: https://www.aclweb.org/anthology/2020.acl-main.335. doi:10.18653/v1/2020.acl-main.335.

[10] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2020) 1234–1240.

[11] Y. Belinkov, L. Màrquez, H. Sajjad, N. Durrani, F. Dalvi, J. Glass, Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks, in: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Asian Federation of Natural Language Processing, Taipei, Taiwan, 2017, pp. 1–10. URL: https://aclanthology.org/I17-1001.

[12] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, M. Baroni, 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 Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2126–2136. URL: https://aclanthology.org/P18-1198. doi:10.18653/v1/P18-1198.

[13] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, N. A. Smith, Linguistic knowledge and

transferability of contextual representations, in: Proceedings of the 2019 Conference of the North, Association for Computational Linguistics, 2019, pp. 1073–1094. URL: http://aclweb.org/anthology/N19-1112. doi:10.18653/v1/N19-1112.

[14] I. Tenney, D. Das, E. Pavlick, BERT rediscovers the classical NLP pipeline, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019, pp. 4593–4601. URL: https://www.aclweb.org/anthology/P19-1452. doi:10.18653/v1/P19-1452.

[15] T. Pimentel, J. Valvoda, R. Hall Maudslay, R. Zmigrod, A. Williams, R. Cotterell, Information-theoretic probing for linguistic structure, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 4609–4622. URL: https://www.aclweb.org/anthology/2020.acl-main.420. doi:10.18653/v1/2020.acl-main.420.

[16] H. Sajjad, N. Durrani, F. Dalvi, F. Alam, A. R. Khan, J. Xu, Analyzing encoded concepts in transformer language models (2022). URL: https://arxiv.org/abs/2206.13289. doi:10.48550/ARXIV.2206.13289, publisher: arXiv Version Number: 1.

[17] F. Dalvi, A. R. Khan, F. Alam, N. Durrani, J. Xu, H. Sajjad, Discovering latent concepts learned in BERT, 2022-05-15. URL: http://arxiv.org/abs/2205.07237. arXiv:2205.07237 [cs].

[18] D. Dessí, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, E. Motta, CS-KG: A large-scale knowledge graph of research entities and claims in computer science, in: U. Sattler, A. Hogan, M. Keet, V. Presutti, J. P. A. Almeida, H. Takeda, P. Monnin, G. Pirrò, C. d'Amato (Eds.), The Semantic Web – ISWC 2022, volume 13489, Springer International Publishing, 2022, pp. 678–696. URL: https://link.springer.com/10.1007/978-3-031-19433-7_39. doi:10.1007/978-3-031-19433-7_39, series Title: Lecture Notes in Computer Science.

[19] I. Augenstein, M. Das, S. Riedel, L. Vikraman, A. McCallum, SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications, 2017-05-02. URL: http://arxiv.org/abs/1704.02853. arXiv:1704.02853 [cs, stat], number: arXiv:1704.02853.

[20] Y. Luan, L. He, M. Ostendorf, H. Hajishirzi, Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction, 2018-08-28. URL: http://arxiv.org/abs/1808.09602. arXiv:1808.09602 [cs].

[21] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, 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), Association for Computational Linguistics, 2019, pp. 3613–3618. URL: https://www.aclweb.org/anthology/D19-1371. doi:10.18653/v1/D19-1371.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019. URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692 [cs].

[23] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019-10-29. URL: http://arxiv.org/abs/1910.13461. arXiv:1910.13461 [cs, stat].

[24] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf,

M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-Art Natural Language Processing (2020) 38–45. doi:10.18653/v1/2020.emnlp-demos.6. arXiv:arXiv:1910.03771v5.

[25] NumPy, 2022. URL: https://numpy.org/doc/stable/, version 1.23.5.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.