Semantic Web 15 (2024) 877–896
DOI 10.3233/SW-233520
IOS Press

A study of concept similarity in Wikidata

Filip Ilievski *, Kartik Shenoy, Hans Chalupsky, Nicholas Klein and Pedro Szekely
Information Sciences Institute, University of Southern California, CA, USA
E-mails: ilievski@isi.edu, kshenoy@isi.edu, hans@isi.edu, nmklein@isi.edu, pszekely@isi.edu

Editor: Harald Sack, FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
Solicited review: Four anonymous reviewers

Abstract. Robust estimation of concept similarity is crucial for applications of AI in the commercial, biomedical, and publishing domains, among others. While the related task of word similarity has been extensively studied, resulting in a wide range of methods, estimating concept similarity between nodes in Wikidata has not been considered so far. In light of the adoption of Wikidata for increasingly complex tasks that rely on similarity, and its unique size, breadth, and crowdsourcing nature, we propose that conceptual similarity should be revisited for the case of Wikidata. In this paper, we study a wide range of representative similarity methods for Wikidata, organized into three categories, and leverage background information for knowledge injection via retrofitting. We measure the impact of retrofitting with different weighted subsets from Wikidata and ProBase. Experiments on three benchmarks show that the best performance is achieved by pairing language models with rich information, whereas the impact of injecting knowledge is most positive on methods that originally do not consider comprehensive information. The performance of retrofitting is conditioned on the selection of high-quality similarity knowledge. A key limitation of this study, similar to prior work lies in the limited size and scope of the similarity benchmarks. While Wikidata provides an unprecedented possibility for a representative evaluation of concept similarity, effectively doing so remains a key challenge.

Keywords: Similarity, Wikidata, retrofitting, knowledge graphs, embeddings

1. Introduction

Large Knowledge Graphs (KGs) support a growing set of real-world knowledge-intensive tasks that rely on the notion of similarity [22,43,99]. Similarity-based search facilitates use cases in a commercial setting, where, e.g., one might search for a dress in a certain price range that is similar to what celebrities wear [99]. The article screening process for systematic reviews of publications can be facilitated by retrieving similar concepts and concept relations [43]. In a biomedical setting, one might integrate knowledge extracted from documents and publications into existing domain ontologies [22]. Workable solutions for such knowledge-intensive tasks like advanced search, information retrieval, and integration rely heavily on robust notions of similarity between concepts, as well as a rich and reliable knowledge about those concepts.

As the largest public KG with billions of statements and a large number of concepts, Wikidata [94] provides knowledge at an unprecedented scale that can intuitively support similarity-driven applications. Recognizing that, for instance, the concept of `natural science (Q7991)` is more similar to `logic (Q8078)` than to `bus (Q5638)` or `plant (Q756)` can be potentially deduced from the concept- or the instance-level knowledge stored in Wikidata. However, as its knowledge has largely been contributed following the Wisdom-of-the-Crowd

*Corresponding author. E-mail: ilievski@isi.edu.

approach [86], Wikidata exhibits notable challenges of deduplication, blurred distinction between classes and instances, and inconsistencies in terms of the modeling of its knowledge [39,74,82]. Because of such challenges, estimating concept similarity in Wikidata is nontrivial. Rather than estimating similarity, prior work on reasoning with Wikidata has focused mostly on methods that solve downstream tasks directly. For instance, a wide range of knowledge graph embedding models has been created to solve the task of link prediction and node classification. Notable early examples are TransE [8] and ComplEx [88], which organize nodes in a geometric space according to their structural links to other nodes. Random walk methods, such as node2vec variants [33,105], leverage the generalizability of language modeling by applying it to graph nodes instead of words. More recently, inspired by the success of contextualized language models such as BERT [20], methods that combine structure and semantics have been devised for these tasks [81,101,102]. Recent work, however, has signaled a lack of generalizability and robustness of these methods [89,104], which further motivates the need to understand the underlying similarity reflected by knowledge graph embedding models.

While estimating similarity between concepts in large KGs is relatively understudied, the related task of word similarity has been very popular in computational linguistics [35,65,78,90]. Early work generally relies on ontology-based similarity metrics [44] that leverage properties of a word in an ontology like WordNet [57]. Ontology-based vector representation models [2,31] leverage the ontological structure in resources like WordNet, e.g., based on random walks, to create word vector representations. More recently, pre-trained word embeddings have been shown to natively capture word similarity at scale [20,51,56]. Early word embeddings have been shown to benefit from knowledge injection [21], again based on lexical resources like WordNet. A variant of the word similarity task, called *WordType* [107], deliberately translates the word similarity task to a concept similarity task by mapping words to nodes in a KG, which provides an opening for these methods to be applied to Wikidata. However, the applicability of existing word similarity tasks and metrics [2,44,48] to KGs such as Wikidata is limited, as these methods have been built for resources like WordNet [57] that are heavily curated and have a clearly distinguishable ontology. These metrics, for instance, do not have inherent mechanisms to benefit from the instance-rich content of Wikidata, nor can they estimate the similarity of multi-word and abstract expressions (e.g., natural science as a branch of science about the natural world). As such, it is important to investigate metrics that are tailored to the size, inconsistency, and open-ended nature of realistic large KGs, such as Wikidata.

In this paper, *we study the ability of ontological and distributional methods to capture the similarity of concepts in Wikidata*. For Wikidata, we assume that concepts correspond to classes that have one or more instances or subclass relations. Aiming to reconcile the rich prior work on word similarity and on KG similarity-related tasks, we design a *framework* that includes representative similarity metrics based on ontological structure, language modeling, knowledge graph embeddings, and their combinations. We experiment with a novel aggregation method, called *TopSim*, that aggregates over different regions in the KG. Example estimations of the similarity between `natural science (Q7991)` and seven other concepts with a representative metric from each of these families are shown in Fig. 1. Besides these metrics, we investigate the impact of knowledge injection, where the idea behind is to adapt the precomputed embeddings to a downstream task by leveraging structured knowledge. We perform knowledge injection with subsets of two large KGs: Wikidata and ProBase [11], and experiment with different relation weighting functions (e.g., based on BERT or ontological metrics). As no similarity benchmarks for Wikidata exist, we adopt three popular word similarity benchmarks by mapping them to Wikidata following the *WordType* setup [107]. This study reveals that many of the existing similarity metrics can be partially adapted for estimating similarity on large-scale knowledge graphs like Wikidata. We observe that retrofitting to Wikidata is typically beneficial, but not to ProBase. This paper concludes with open questions regarding novel metrics that focus on the wealth of instance knowledge in Wikidata, a meaningful evaluation of similarity between two Wikidata concepts, and the need for novel benchmarks that support the evaluation of concept similarity in KGs.

In summary, we make the following contributions:

1. We motivate the task of estimating the similarity between pairs of concepts in Wikidata. While prior work has considered word similarity and well-curated resources like WordNet, this is the first paper that addresses conceptual similarity in large and relatively noisy sources such as Wikidata (Section 2).
2. We present a computational framework that includes three families of models, publicly available knowledge for self-supervision, and a retrofitting method that leverages this knowledge to customize the models (Sec-

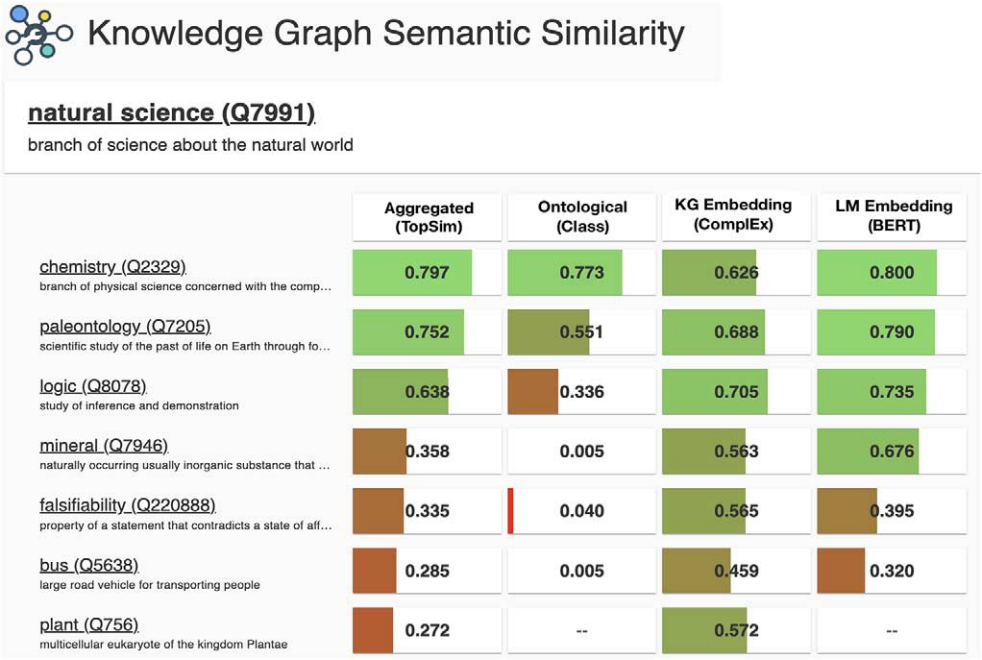

Fig. 1. Example of similarity judgments between natural science and seven other concepts in Wikidata.

tion 3). To our knowledge, we are the first to propose the TopSim metric that aggregates over KG regions, and we are the first to retrofit similarity estimations based on knowledge in Wikidata and ProBase.

3. We adapt three benchmarks to Wikidata (Section 4) and perform extensive experiments that investigate the comparative success of different models and their combinations, the impact of retrofitting, and the impact of various background knowledge sets (Section 5).

The entire code and pointers to the data supporting our experiments can be found on GitHub: https://github.com/usc-isi-i2/wd-similarity.

2. Capturing similarity of concepts

In this section, we provide background information from cognitive studies on similarity and put the problem of automatically estimating conceptual similarity in the context of prior work in the fields of Computational Linguistics and Semantic Web.

2.1. Background

Conceptual similarity is a central theoretical construct in psychology, facilitating the transfer of a situation to an original training context [54,62]. Tversky [91] poses that the *literal similarity* between two objects A and B is proportional to the intersection of their features and inversely proportional to the features that differ ($A - B$ and $B - A$). In other words, A and B are literally similar if their set of shared features is relatively larger than the non-shared ones. Here, features include both attributes and relational predicates. This differs from *analogy*, where only relational predicates are shared (e.g., atom – solar system), and from *mere appearance*, where only the attributes are shared and not the relationships (e.g., moon – coin) [27]. When A and B do not have any attribute nor relational overlap (e.g., coffee and solar system), this is called anomaly [27].

Gentner and Markman [29] argue that similarity is like an analogy, in the sense that both rely on the alignment between the two compared objects or domains. When provided a pair of similar concepts, for example, hotel-

motel, people align them based on their shared properties (e.g, used for accommodation), and are able to easily point out differences (e.g., hotels are in cities, motels are along highways). The authors discuss that it makes no sense to talk about differences in the absence of a meaningful alignment (e.g., kitten – magazine). Which features are or should be considered when computing similarity? According to [60], the relative importance of a feature depends on the stimulus task and the context. This flexibility of the estimation of similarity led to criticism, which argues that the set of features that are being compared is seemingly arbitrary [32]. Yet, the relatively high inter-annotator agreement across human subjects when judging similarity indicates systematicity in the human judgments of similarity [54]. In addition, some of the variations across human subjects can be explained with phenomena like selective learning [83], developmental changes [28], and knowledge and expertise [13]. Similarity judgments are also known to be impacted by the context of the task [79], i.e., the features activated depend on the object we compare against; as well as the direction of the comparison: people tend to rate the similarity of North Korea to China higher than the reverse [54].

One can distinguish three measures of similarity: indirect, direct, and theoretical [54]. An example for indirect similarity comparison is asking human participants to identify potentially confusable stimuli, such as judging whether an object has been observed before. Similarity can be measured directly, by rating the similarity of stimuli on a numeric scale. The theoretical similarity is observed as a component in human cognition, for instance, when participants categorize an item by comparing its fit in various categories.

In this paper, we consider the task of literal similarity between two concepts. Given two concept nodes, c_1 and c_2 described in a KG G, a system is asked to provide a numeric pairwise similarity score $sim(c_1, c_2)$ that would mimic the average judgment provided by human subjects.

2.2. Related work

Word similarity metrics Computational Linguistics research has studied the extent to which two words are similar or related. Here, similarity follows the notion of literal similarity in psycholinguistics, while relatedness is a broader notion that indicates that two words tend to appear in the same topical context [9,65]. Literally similar words are found nearby in an *is-a* hierarchy, whereas related words connect through another relation, e.g., causality or part-whole [65]. In practice, the similarity between two words is evaluated by comparing the human scores to the algorithmic scores. Attaching meaning to the absolute scores is difficult, thus it is common to consider the scores across pairs relative to each other, and compare the pair similarity ordering between the algorithm and humans.

Lastra-Díaz et al. [48] survey a wide range of methods for word similarity, and they categorize the methods into two large families: ontology-based semantic similarity and distributional features captured with word embeddings. The ontology-based metrics are further divided into topological measures, gloss-based measures, and ontology-based vector models. Word embedding metrics are divided into two groups: text-based word embeddings and word embedding computation of ontologies and distributional models. Multiple ways to use ontologies and distributional models together exist, including joint approach [69,98], injection [21,59], and embedding combination [30]. Our framework for conceptual similarity in Wikidata is partially aligned with the categorization of [48] for word similarity. Namely, we include topological measures, as well as ontology-based vector models in the form of graph embeddings. Within word embeddings, we specifically focus on language model embeddings, which have become superior in the meantime. Rather than only combining knowledge and word embeddings, we promote the knowledge injection component to an independent component in our framework that enriches both embeddings of language models and knowledge graph nodes, in an attempt to emphasize the extensive knowledge available in external sources like Wikidata. We also include embedding combinations as a separate framework component, where we contribute a novel combination approach, called TopSim. Following common practice in the word similarity tasks, we assume that the similarity of two concepts can be measured on a continuous numeric scale. We compare the relative order of the machine similarity for a dataset against human judgments.

Concept similarity metrics Algorithmic measures that capture the similarity between two concepts can be classified broadly into corpus-based and knowledge-based metrics [35]. Corpus-based semantic metrics are based on text analysis and typically rely on the distributional hypothesis that concept meanings can be inferred based on their co-occurrence in language [6]. Language structure comes from two aspects: paradigmatic and syntagmatic.

According to the paradigmatic view, linguistic symbols are regarded as paradigms that are members of a specific group (e.g., nouns), whereas syntagmatic relationships are formed between surface symbols (e.g., words) to form a syntagm and define the meaning of a sentence. While set-based [7] and probabilistic [14] methods exist, most distributional metrics are geometric, including Latent Semantic Analysis [47], Explicit Semantic Analysis [25], and Hyperspace Analogue to Language [52]. State-of-the-art distributional measures of similarity are based on large-scale language models, such as Word2Vec [56], GloVe [67], BERT [20], and RoBERTa [51]. Knowledge-based metrics rely on the analysis of ontological structures, which formally indicate how concepts (i.e., words grounded in the ontology) relate to each other. Metrics based on graph structure estimate the similarity as a function of the degree of interconnection between concepts, captured through shortest paths [75], ontology depth [49,77], concept specificity [66,100], and concept density [3]. The main idea of the feature-based metrics is to represent concepts as sets of features and compute similarity by comparing these sets [17,53]. Information theoretical methods are typically based on the Information Content (IC) of the concepts or their common ancestors [44,73,77], where IC is typically a proportion of all instances that belong to a concept. Hybrid measures combine aspects from multiple metric categories, e.g., depth and density with information content [44]. In recent years, various graph embeddings have been proposed [8,88], which can be used to estimate concept similarity based on cosine distance in vector space. We design our framework to consolidate these corpus- and knowledge-based metrics. We include corpus metrics based on LM embeddings, and two families of knowledge-based metrics: ontology-based topological metrics and KG embedding metrics.

Concept similarity in Linked Data sources Similarity allows for direct comparison between two KG node representations, thus facilitating the matching of ontological schema across multiple knowledge sources [70]. In [12], the authors propose a LOD-based similarity measure based on the combination of ontological, classification, and property dimensions of knowledge. This metric has been used on a set of RDF graphs, centered around DBpedia, and evaluated on three-word similarity benchmarks mapped to DBpedia. Zhu and Iglesias [107] formalize the tasks of Word-Noun, Word-Graph, and Word-Type, all of which adapt traditional word similarity metrics to concepts in Linked Open Data sources such as DBpedia. They propose a novel metric that combines the semantic network structure between concepts with the information content of the concepts. In follow-up work, Alkhamees et al. [4] devise a metric that exploits the information content of the least common consumer of two concepts. Our work is complementary to prior efforts that propose novel similarity metrics for the similarity of the concepts in DBpedia, as we motivate the task of estimating the similarity of pairs of concepts in Wikidata, a KG with different properties compared to WordNet and DBpedia. We address this task with a computational framework that includes metrics based on information content and graph structures, but also metrics based on language models that focus on the literal content present in Wikidata.

Entity similarity in Linked Data sources REW OrD [72] is a method to estimate relatedness between two entities, based on predicate frequency – inverse triple frequency (PF/ITF). It has been applied on DBpedia and LinkedMDB. The information content metric, called Partitioned Information Content Semantic Similarity (PICSS) [55], aims to give more importance to significant relations. Caballero and Hogan [10] propose four metrics for global node similarity in Wikidata. They also release two Wikidata benchmarks for entity similarity: movies and music albums. Most specific similarity query (MSSQ) [71] is a similarity algorithm that can estimate similarity of two entities in a single RDF graph. The evaluation in this work shows that an approximation of MSSQ scales well, without an indication of the extent to which the computed similarities correspond to human judgments. In [63] the author proposed a Linked Data Semantic Distance (LDSD) which relies on direct and indirect relationships between two DBpedia resources. The distance measure was employed in a music recommendation system [64]. More recently, Wang et al. [97] propose a matrix factorization method that enhances recommendation with an LOD-based semantic similarity measure. Long-tail entities are explicitly considered by the collaborative filtering method in [61]. In [37], the similarity is used to judge whether two representations refer to the same long-tail entity. In order to account for the large sparsity of long-tail entity knowledge, the representations include probabilistic information, based on models over instance-level knowledge in Wikidata. Rather than focusing on the similarity between KG entities, we study the effect of adapting various methods that capture conceptual similarity to large KGs like Wikidata.

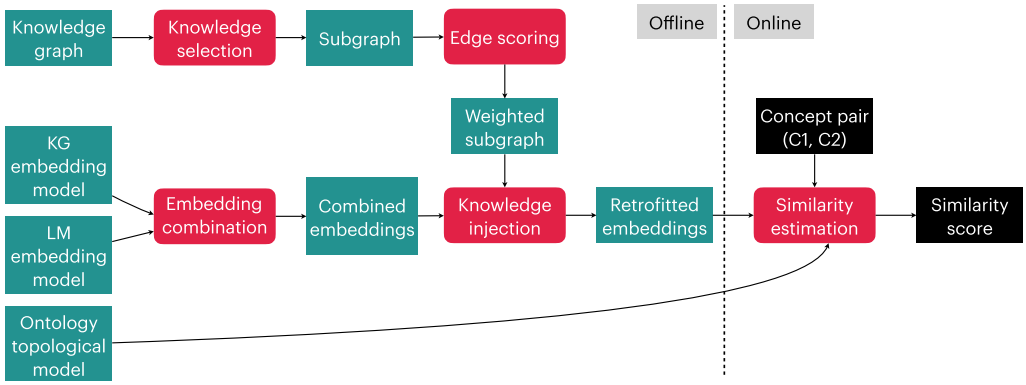

Fig. 2. Overview of our similarity estimation framework.

3. Framework for estimating similarity

Our framework for estimating similarity is visually depicted in Fig. 2. We use graph embedding and text embedding models, as well as ontology-based metrics, as initial similarity estimators. We also concatenate the embeddings in order to combine their scores. We use retrofitting as a knowledge injection mechanism to further tune the individual or the combined embedding models, through distant supervision over millions of weighted pairs extracted automatically from large-scale knowledge graphs. For a given concept pair from Wikidata, the similarity scores generated by the retrofitted embedding models can be combined with the scores by the ontology-based models. We next describe the individual components of our similarity framework.

3.1. Similarity models

We distinguish between similarity models based on KG embeddings, language models, and ontology-based topological metrics. We employ representative methods from each category, as well as aggregate methods that combine them in intuitive ways. The goal of our framework is to evaluate a representative sample of metrics and their combinations for the novel task of estimating similarity between Wikidata nodes. To the extent possible, we follow the terminology of [48] and we adapt methods and ideas from prior work on word similarity and concept similarity with WordNet and DBpedia. We leave it to future work to perform an exhaustive evaluation of the wide range of metrics for word and concept similarity developed in the past, e.g., for WordNet [35,48], within our framework.

Knowledge Graph Embedding (KGE) metrics KGE models have been popular for graph-based tasks like link prediction and node classification [96], which rely on the similarity between nodes or graph structures. The KGE models are analogous to the Ontology-based Vector Models in [48]. We experiment with four KG embedding models, which can be divided into translation, tensor factorization, and random walk models.

We adopt popular models based on translation, *TransE* [8], and tensor factorization, *ComplEx* [88]. We select TransE and ComplEx because they have been commonly applied for representation learning of KGs and for KG prediction tasks that relate to similarity (e.g., link prediction) [16]. Moreover, these two models are relatively simple and intuitive, which enables an understanding of their behavior.[1] TransE represents graph nodes and relations in the same space. Given a KG triple (h, r, t), the relation r acts as a translation vector that connects the nodes h to the nodes t. The intuition behind TransE's translation comes from distributed word representations that capture linguistic regularities in the form of proportional analogies, e.g., (keyboard, button) is analogous to (office, chair), as both pairs are connected with the relation *part-of*. The embeddings in TransE are then optimized to minimize the distance between $h+t$ and t in an embedding space. Our tensor factorization model, ComplEx, introduces complex-valued embeddings that allow it to better model asymmetric relations. The main novelty of ComplEx is that its node

[1] We note that these models may not achieve state-of-the-art across KG tasks and cannot encode semantic information about literals. Including state-of-the-art methods, like RotatE [85] and JointE [106], and literal-aware methods, like LiteralE [46] and KGA [95], is a key future pursuit.

and relation embeddings lie in a complex, rather than real, embedding space. Its loss function is designed to assign different scores to asymmetric relations depending on the order of the nodes involved in the relation. For further details on these models, we direct the reader to the survey by Wang et al. [96]. TransE, ComplEx, and related KGE models can be readily applied to estimate the similarity between any two nodes (concepts or entities) in Wikidata. Once the embeddings by a model are computed, we estimate the similarity between two nodes through the cosine similarity between their KGEs. As Wikidata is not based on a well-founded formal ontology and also does not explicitly separate TBox and Abox, we train translation and tensor factorization embeddings over the entire set of node-node relationships in Wikidata as is commonly done. We do not perform any pre-processing or pre-selection, i.e., we treat all relationships in Wikidata as one large KG.[2]

For random walk models, we adopt *DeepWalk* [68]. Analogous to our choice of translation models, we choose DeepWalk given its simplicity and retained popularity for KG prediction tasks to this day. DeepWalk is a method that uses skip-gram to train over randomly generated sequences of nodes ("random walks") in a KG, and generates the vector of nodes involved in that sequence. This model is analogous to skip-gram models in natural language, like word2vec [56]. DeepWalk assumes that distributed representations of nodes can be learned through a stream of relatively short walks, which capture neighborhood similarity and community membership, according to the homophily principle [105]. We also experiment with a variant of Deepwalk, which we call *S-DeepWalk*, originally proposed in [105], which is designed to capture the structural similarity between nodes. For example, two professors may have a high structural similarity if they play analogous roles in their social networks, e.g., connecting to many students or working for an educational institution. The input graph for generating walks in S-Deepwalk is, thus, constructed by connecting the nodes that have neighbors from similar classes in the original KG. For Wikidata, we use class membership as indicated by its instance-of property (P31). Identical to the translation models, we estimate similarity with the random walk models by computing cosine similarity of their resulting embeddings.

Language Model Embedding (LME) metrics We use Transformer LMs based on BERT [20] to represent textual information associated with a node in the graph, which broadly correspond to the Word Embeddings category in [48]. BERT [20] is a pre-trained Transformer network [92] that has been optimized for the tasks of language modeling and next sentence prediction. In BERT, a Multi-head attention over 12 (base-model) or 24 layers (large-model) is applied and the output is passed to a simple regression function to derive the final prediction. In this work, we use the sentence-transformer [76] method, which uses use siamese and triplet network structures to derive semantically meaningful sentence embeddings. We compare the resulting embeddings using cosine similarity. To adapt BERT-based models to the task of estimating similarity between concepts in Wikidata, it is required to transcribe the information about a node in natural language. As describing a KG node with a representative text is an active research area [1,34], we experiment with two input variants: using human-generated *abstracts* and machine-generated *lexicalizations* based on Wikidata.

Our *Abstract* model is based on the abstracts found in the DBpedia KG. We map these abstracts to Wikidata via the *sitelinks* dataset, which is available via our GitHub repository. For the nodes that could not be automatically mapped to DBpedia abstracts, we use a fallback strategy of concatenating their Wikidata label and description. We use the first sentences of the aligned abstracts, because we observe that using the first sentence performs on par with using the entire abstract. In our *Lexicalization* variant, we automatically generate a node description based on five optional kinds of information, which we illustrate for the concept of cat (Q146): 1) node labels (e.g., "house cat"), 2) node descriptions (e.g., "domesticated feline"), 3) is-a values (e.g., cat is an organism known by a name), 4) has properties (e.g., cat has a heart rate and life expectancy), and 5) property values (e.g., cat is studied by felinology). Each of the five elements is optional, and requires further specification in terms of which properties should be used for which element. The selected pieces of information are seamlessly combined with a template.[3] An example lexicalization for this example would be "House cat, domesticated feline, is an organism known by a name, has a heart rate and life expectancy, and is studied by felinology".

[2] As a side note, understanding the quality of KG embeddings trained on large KGs for similarity tasks is a timely pursuit, as KG embeddings so far have mainly been trained on relatively small graphs and evaluated on the task of link prediction [26,95].

[3] For more information, we refer the reader to the documentation page: https://kgtk.readthedocs.io/en/latest/transform/lexicalize/.

Ontology-based Topological (OT) metrics OT metrics use features derived from the structure (topology) of the underlying base ontology [48]. We adopt two OT metrics. *Jiang Conrath* [44] is an information-theoretic distance metric that combines path-based features with information content. The Jiang Conrath metric leverages the information content of the least common subsumer which is given by $jc(c1, c2) = 2 * \log p(mss(c1, c2)) - (\log p(c1) + \log p(c2))$. Here, $mss(c1, c2)$ is the most specific subsumer, whereas $p(c)$ is the normalized probability that a particular concept c is of type C. We normalize Jiang Conrath distances onto a $[0 \ldots 1]$ similarity measure by dividing by the largest possible distance between $c1$ and $c2$ through the root node in the ontology.

Our second metric, which we dub *Class similarity*, has been inspired by methods that use inverse document frequency (IDF) for string matching for table linking [87]. While, to our knowledge, the idea of IDF has not been applied directly to the task of word and concept similarity, related metrics based on information content have been relatively popular [48]. Our formulation of this method for concept similarity consists of four steps (we illustrate them for $c1$ being dog and $c2$ being lion):

1. We compute the set of IsA parents for both concepts based on their instanceOf (P31) and subclassOf relations (P279). Formally: $IsA(c1) = P31/P279^*(c1)$ and $IsA(c2) = P31/P279^*(c2)$. For our example, $IsA(dog) = [pet, mammal, animal]$ and $IsA(lion) = [big\ cat, Felidae, mammal, animal]$.
2. We compute the common parents for both concepts, formally, $common(c1, c2) = IsA(c1) \cap IsA(c2)$. For our example, $common(c1, c2) = [mammal, animal]$.
3. We compute the total *class_sim* score as a sum of the IDF scores of the common classes, namely, $class\_sim = \sum IDF(c)$, where $c \in common(c1, c2)$. We compute the IDF of a class c as a ratio between the number of instances of the class c and the total number of Qnodes in the KG. For our example, let's assume that $IDF(mammal) = 0.7$ and $IDF(animal) = 0.18$. Then $class\_sim(dog, lion) = 0.88$.

3.2. Embedding combination

Composite embeddings Considering that KG and LM embeddings may provide complementary insights for similarity [40], we create two composite embeddings. *Composite-all* combines all our embedding models: two translation models (TransE and ComplEx), two random-walk models (Deepwalk, S-Deepwalk), and two LMs (Abstract and Lexicalize). *Composite-best* combines the best KG and the best LM embeddings. For the composite embeddings, we estimate cosine similarity in the same way as with the individual models.

TopSim Similarity functions vary significantly in the kind of similarity they capture. They are also generally better at ranking within the high-similarity region, for example, whether *cat* is more similar to *dog* than to *horse*, but then become more random in the long tail, such as whether *cat* is more similar to *car* than to *house*. For this reason, we develop TopSim,[4] which aggregates different base similarity measures into a combined measure for more robust performance and ranking. In addition to aggregating individual measures, TopSim averages over a whole region of high-similarity KG nodes for added robustness. Next, we describe TopSim in detail.

TopSim is an aggregation framework that can be instantiated with an arbitrary set of measures. Let $sim_i(x, y)$ be a set of N base similarity measures that compute a similarity between two KG nodes x and y. We define a basic combination measure as a weighted average of the chosen base measures:

$$csim(x, y) = \frac{\sum_{i=1}^{N} w_i \times sim_i(x, y)}{\sum_{i=1}^{N} w_i}$$

Let $cand_j(x)$ be a set of M candidate generation functions to compute potential high similarity regions for any node. Candidate regions serve two basic purposes: (1) they make the computation of high-similarity regions more efficient by not having to compare a node to all other nodes in the KG, and (2) they reduce the noise that comes from comparing a node to everything else, since similarity functions are generally not very good at ranking outside the

[4]This is not to be confused with a measure of the same name but different semantics described in [50].

high similarity region. Candidate generation functions can be implemented by looking for the most-similar nodes of some embedding-based measures, or by doing a constrained walk of the ontology, for example.

For each node x and y we compute a ranked top-similarity region *tsim* of size T as follows:

$$tsim*(x) = \left(c_k \in \bigcup_{j=1}^{C} cand_j(x) \middle| csim(x, c_k) \geqslant csim(x, c_{k+1}) \right)$$

$$tsim(x) = \left(c_1, \ldots, c_T | c_i \in tsim*(x) \right)$$

This means we take the union of all candidate sets $cand_j(x)$, compute $csim(x, c)$ for each candidate, order them by their similarity to node x and then take the top-T elements of that set as the result of $tsim(x)$. Finally, given the two top-similarity regions $tsim(x)$ and $tsim(y)$, we compute $topsim(x, y)$ as a weighted average of the cross-product of similarities between the two regions:

$$topsim(x, y) = \min\left(\frac{csim(x, y) + \sum_{cy_i \in tsim(y)} csim(x, cy_i) + \sum_{cx_i \in tsim(x)} csim(y, cx_i)}{1 + \sum_{cx_i \in tsim(x)} csim(x, cx_i) + \sum_{cy_i \in tsim(y)} csim(y, cy_i)}, 1 \right)$$

The intuition here is that instead of just computing $csim(x, y)$, which would simply average the base similarity measures for the two nodes, we additionally aggregate similarities between a node and all the candidates in the top-similarity region of the other node, which is more robust than looking at individual nodes alone.

3.3. Knowledge injection

Our KG and LM embedding models, as well as their combinations, can be expected to capture a wide range of rich information about concepts. KGEs will capture information about concepts in relation to their instances in Wikidata, whereas LM embeddings will connect a portion of the graph information with the rich background knowledge that LM models have acquired during pretraining. However, none of these models are directly intended for the task of concept similarity over Wikidata. Their application to estimate similarity is relatively speaking in a "zero-shot" manner. In this section we experiment with retrofitting, a technique for knowledge injection that is aimed to tune the models to the task of concept similarity in Wikidata. To support retrofitting, we experiment with subsets from two large knowledge graphs: Wikidata and ProBase. We select Wikidata to understand whether directly tuning on the graph used for prediction can improve model performance. We include ProBase because of its rich coverage of is-a knowledge, which is essential for estimating model similarity.

Retrofitting We use the retrofitting technique proposed by Faruqui et al. [21] as a straightforward and intuitive knowledge injection method. Retrofitting iteratively updates node embeddings in order to bring them closer in accordance to their connections in an external dataset. Retrofitting is a weighted model, i.e., the impact of the neighbors is not constant and can be flexibly specified. Given a node with an embedding q_i, and n neighbours, where the jth neighbour with weight β_j has an embedding q_j, the retrofitted embedding $\hat{q}_i$ is computed as follows:

$$\hat{q}_i = \frac{q_i \times n^k + \sum_{j=1}^{n} q_j \times \beta_j}{n^k + \sum_{j=1}^{n} \beta_j}$$

The parameter k dictates how much the original embedding will be changed based on its neighbors. Higher k values result in higher preservation of the original embedding. While retrofitting has been introduced for lexical resources like WordNet [57] and FrameNet [5], in this paper, we adapt retrofitting to tune the KGE and LM embeddings with large-scale background knowledge. Namely, we tune the original embeddings by knowledge injection from two large KGs: Wikidata and ProBase.

Wikidata We derive three datasets from Wikidata's subclass-of (P279) ontology. First, *WD-child-parent* consists of subclass-of relations whose nodes have non-empty and non-identical labels and descriptions, and both nodes have at least 10 P279 descendants. If more than 500 children are present for a parent, we randomly sample 500 of them. This yields 304k relations in total. Second, *WD-sibling* is a dataset with sibling relations between two nodes in WD-child-parent that share the same immediate parent and have a non-identical description. WD-sibling contains 785k relations, e.g., dog – sibling – cat. Third, we construct *WD-all* as a union of WD-child-parent and WD-sibling. From a total of 472,563 Qnodes in *WD-all*, there are 241,699 (51%) Qnodes that have a short abstract, whereas, for the remaining 49%, we use a fallback strategy of concatenating their Wikidata label and description (*Label* + *Desc*).

We define three weighing methods for the generated pairs from these datasets: (1) *constant* weighting value of 1; (2) *class* similarity between the two nodes (using the Class metric described in Section 3.1); and (3) *cosine* similarity between the concatenated labels and descriptions of the two nodes. For WD-child-parent pairs, we compute cosine similarity of the LM embeddings of their labels and descriptions, whereas for the WD-sibling pairs, we compute cosine similarity between the LM embeddings of their sentences. Both sentences follow the template `{Label}, {Description}, is {Parent}`. We use the absolute cosine values as similarities, formally $sim(c1, c2) = |cosine(c1, c2)|$. We focus our experiments on cosine similarity as a weighting function, because we observed empirically that it consistently performs better or comparable to the other two weighting functions. We take its absolute value because we interpret negative similarity as an indicator of antonymy rather than dissimilarity.

ProBase [11] has child-parent pairs associated with a count of occurrences (relations) in textual sources. Based on an exact matching strategy, we aligned 1.6 M (4.79 %) of these pairs to Qnodes in Wikidata. If multiple Qnodes are retrieved, we choose the Qnode with the lowest numeric ID. We opt for this strategy as a lightweight heuristic to obtain popular entities, as they have typically been added to Wikidata early and their IDs are typically low [15]. Meanwhile, popularity has been shown to be a strong and lightweight entity linking baseline [15,41]. The number of relations in this subset range from 1 to 35,167 per subject node. As most nodes have a small number of relations, we scale the edge weight based on the following equation: $\beta = s \times (1 + \frac{\log(no\_of\_relations)}{\log(\max(no\_of\_relations))})$.

4. Experimental setup

4.1. Benchmarks and metrics

This paper is based on the premise that the similarity between Wikidata concepts should be evaluated with large-scale benchmarks. However, while we were able to infer large datasets for data augmentation at training time, large-scale similarity evaluation datasets for Wikidata are not available. As a proxy, we turn to the existing word similarity benchmarks [48]. Adapting word similarity datasets to Wikidata requires (dominantly) manual mapping of words into Qnodes, which is laborious and expensive. Therefore, we select three popular word similarity benchmarks for this paper and map them to Wikidata, resulting in the novel benchmarks WD-WordSim353, WD-RG65, and WD-MC30. With these benchmarks, we follow the *Word-Type* task formulation [107]. In the Word-Type setup, words are mapped to concepts in a knowledge graph, and word pairs that cannot be mapped to a KG are left out of the benchmark.

WD-WordSim353 is derived based on the popular word similarity dataset WordSim-353 [23].[5] WordSim-353 contains 353 pairs of English words (334 without duplicate pairs), along with human similarity scores. Words in WordSim-353 were mapped to Wikidata in a semi-automatic manner, by first using text search to obtain top Wikidata candidates for a word, followed by a manual validation or reranking of the top result performed by a single human annotator. As we observed that the original scores often conflate the notions of semantic similarity (*car-bike*) with relatedness (*car-wheel*), we re-annotated the dataset with numeric scores between 1 and 4: using 1 for (near-) identity; 2 for cases where two entities are partially substitutable, one is a slight specification of the other,

[5] http://www.gabrilovich.com/resources/data/wordsim353/wordsim353.html

Table 1
Statistics of our evaluation benchmarks

Benchmark	#pairs	#unique concepts
WD-WordSim353	334	420
WD-RG65	34	31
WD-MC30	16	23

or they are close siblings of the same category; 3 for pairs that are related by one of the following relations: distant inheritance, location, utility/capability, part-whole, antonymy, or domain; and 4 for unrelated pairs. Five researchers participated in this annotation. After an initial trial phase that allowed us to set the annotation guidelines, in the actual annotation, we observe a Fleiss' kappa [24] agreement of 0.542, which is a moderate agreement. We opt for averaging the judgments rather than resolving the disagreements. This is because our annotation aims to capture the immediate intuition of humans when presented with a pair to judge. We believe that similarity judgments should not be negotiated, as this would take away the intuitive nature of the task. Rather, we believe that disagreements may be a useful signal to indicate the implicit context that the annotator considers subconsciously, however, pursuing this direction is out of the scope of our paper.

WD-RG65 Our second benchmark, WD-RG65, is based on the DBpedia disambiguation [12] of the RG-65 benchmark [80]. The original paper [12] indicates that 54 of the pairs in RG-65 were mapped to DBpedia, without providing specific details about the disambiguation procedure. We derived WD-RG65 from its DBpedia version by using sitelinks data. After linking these pairs to Wikidata, our benchmark WD-RG65 consists of 34 concept pairs featuring 32 unique concepts.

WD-MC30 is a benchmark that is based on the DBpedia disambiguation [12] of the MC-30 benchmark [58]. The original paper [12] states that 25 of the pairs were mapped to DBpedia, and it does not provide specific details about the disambiguation procedure of these nodes. Analogous to WD-RG65, we translated the DBpedia identifiers to Wikidata by using sitelinks data. Our resulting benchmark, WD-MC30, consists of 16 concept pairs featuring 23 unique concepts.

Statistics about the three benchmarks can be found in Table 1. We evaluate using three metrics to measure correspondence with human annotations: Kendall-Tau (KT), Spearman rank (SR), and Root Mean Square Error (RMSE). We make the resulting benchmarks available for future evaluations through our project's GitHub page.

4.2. Modeling and implementation details

Individual models To train the KGE models, we use the DWD version of Wikidata based on its February 15th, 2021 dump [95]. We extract its relationships between two Qnodes, i.e., we preserve the subset of this data where both the subject and the object are Qnodes. The statistics of the resulting graph are shown in Table 2. We note that the node degrees has a high variance across the graph, with a median of 3, a mean of 8.5, and a standard deviation of 2285. Meanwhile, the graph has a high Qnode-to-property ratio. To understand the distribution of the properties further, we plot their frequency distribution in Fig. 3. Similar to nodes, we observe a high imbalance across properties. Certain properties, like instance-of (P31), country (P17), and located in the administrative territorial entity (P131), are very common, with tens of millions of statements. Meanwhile, most of the properties participate in less than 10,000 relationships. While we expect that the skewness of the data may have an impact on the model performance and its ability to capture the KG information [42], experimentation with data balancing and sampling strategies is beyond the scope of this paper.

For Lexicalization, we use the BERT embeddings provided by [45], which have been shown to work relatively well for entity similarity. These embeddings include the following Wikidata properties: P31 (instance of), P279 (subclass of), P106 (occupation), P39 (position held), P1382 (partially coincident with), P373 (Commons Category), and P452 (industry). We include two baseline versions of Lexicalization that use restricted inputs: *Label*, which considers only the English label of a concept as found in Wikidata; and *Label* + *Desc*, which considers a concatenation between a node label and its description in English as given in Wikidata. We note that these descriptions are different from the abstracts from DBpedia, though the two may overlap partially. For instance, the DBpedia abstract

Table 2
Statistics of our subset of Wikidata, which is used as a basis to train the KGE models

Statistic	# Qnodes	# Properties	# Relationships	Mean Degree	Median degree	Stdev Degree
Value	53,002,670	1,368	225,893,116	8.52	3.0	2285.05

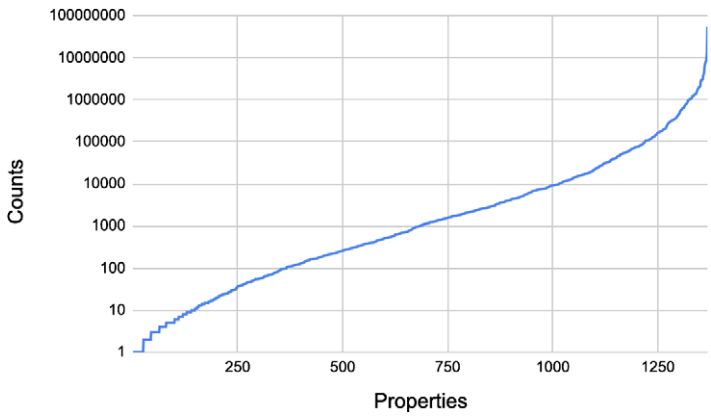

Fig. 3. Frequency distribution of properties in the data used to train the KGE models.

for universe is "The universe (Latin: universus) is all of space and time and their contents, including planets, stars, galaxies, and all other forms of matter and energy.", whereas the Wikidata description is "totality consisting of space, time, matter and energy". For Label, Label + Desc, and Abstract we create novel embeddings with DistilRoberta.[6] For the Class similarity model, IsA relations are computed as a transitive closure over both the subclass-of (P279) and the instance-of (P31) relations. For Jiang Conrath, instance counts are computed using the following properties: P31 (instance of), P39 (position held), P106 (occupation) and transitive P279 (subclass of). The reason for using positions and occupations in addition to P31 is that more descriptive classes (e.g., actor) are usually not linked via P31 which generally only points to Q5 (human) in those cases. The dimensions of the different embeddings are as follows: ComplEx – 100, TransE – 100, Deepwalk – 200, S-Deepwalk – 200, Lexicalize – 1024, Abstract – 768, Label – 768, and Label + Desc – 768.

Model combinations When creating composite embeddings, we use tSNE to reduce the dimensions of the individual models to the same size. For the results reported in this paper we compute TopSim using Class, Jiang Conrath, ComplEx, TransE, and Lexicalization as the $N = 5$ base similarity measures and combine them in an unweighted simple average. We use the top-100 similar nodes based on ComplEx and a Node2Vec embedding-based similarity measure to compute candidate regions, plus an ontology-based candidate function that first goes up one step in the class hierarchy from a source node and then down one step to enumerate ontology-based neighbors. For the final aggregation, we compute similarities over the top $T = 5$ most similar nodes to each of x and y.

Knowledge injection details We experiment with three values for the retrofitting variable k: 0.5, 1, and 2. We generally observe best results with $k = 2$ and 2 iterations of retrofitting, and we present results for this configuration. We use DistilRoberta embeddings [76] to compute cosine similarity as a weighting function. We use the DWD version of Wikidata, whose statistics are given in Table 2. We use the latest version of ProBase, which includes 33M triples.[7] We use $s = 0.5$ as a scaling factor for ProBase.

Implementation details We use the KGTK [36] toolkit to lexicalize a node, subset the graphs, and create various graph and language model-based embeddings. We use scikit-learn for supervised learning. We use KGTK's similarity API [40] to obtain scores for the metrics Class, Jiang Conrath, and TopSim.[8]

[6] https://huggingface.co/sentence-transformers/all-distilroberta-v1
[7] https://concept.research.microsoft.com/Home/Download, accessed on January 21st, 2022.
[8] https://kgtk.isi.edu/similarity_api

Table 3

Correlation scores for the raw methods and combinations that we have, for each of the benchmarks: Kendall-Tau (KT), Spearman rank (SR), and Root Mean Square Error (RMSE). Best values per column are marked in bold

Type	Model	WD-WordSim353				WD-RG65			WD-MC30		
		Coverage	KT	SR	RMSE	KT	SR	RMSE	KT	SR	RMSE
KGE	TransE	334	0.22	0.305	0.699	0.182	0.301	0.793	0.133	0.244	0.87
	ComplEx	334	0.208	0.294	0.81	0.161	0.274	0.748	0.25	0.426	0.763
	Deepwalk	327	0.281	0.392	0.731	0.238	0.322	0.741	0.291	0.422	0.683
	S-Deepwalk	226	0.042	0.055	0.916	0.03	0.054	1.17	−0.143	−0.201	1.177
LME	Abstract	334	**0.523**	**0.697**	**0.523**	**0.518**	**0.662**	**0.592**	**0.567**	**0.753**	**0.568**
	Lexicalization	334	0.374	0.512	1.031	0.408	0.581	0.734	0.45	0.597	0.799
	Label	334	0.041	0.059	0.732	−0.032	−0.047	0.909	0.167	0.218	0.948
	Label + Desc	334	0.368	0.508	0.646	0.132	0.219	0.897	0.25	0.388	0.937
OT	Jiang Conrath	334	0.28	0.393	0.725	−0.065	−0.095	1.03	0.076	0.1	1.074
	Class	334	0.319	0.441	0.741	0	−0.031	1.054	−0.059	−0.091	1.14
Comb.	Composite-All	334	0.437	0.587	0.707	0.304	0.432	0.882	0.25	0.409	0.97
	Composite-Best	334	0.488	0.654	0.572	0.408	0.516	0.718	0.45	0.585	0.729
	TopSim	334	0.382	0.517	0.703	0.257	0.37	0.660	0.217	0.324	0.764

5. Results

5.1. How well do different algorithms and combinations capture semantic similarity?

As can be seen from Table 3, the Abstract model performs best among all language model variants, and overall. It outperforms the other LMs because DBpedia's abstracts, being written by humans, contain information that is more comprehensive and tailored to concepts than automatically lexicalized knowledge in Wikidata. The Lexicalization model outperforms the other simpler baselines, but it may be improved further by a dynamic selection of properties, e.g., through profiling [45]. Language models perform worst when they consider labels only, which can be expected because node labels contain the least information. Adding a description to labels yields a notable improvement, and the Lexicalization method further improves upon Labels + desc. These results together show that the exact kind and amount of information fed to language models matters strongly for estimating similarity.

The graph embedding methods each focus on abstracting the rich information available in Wikidata. Among these methods, the Deepwalk embeddings perform the best. These methods are consistently outperformed by the Lexicalization and Abstract methods, suggesting that the graph embeddings' wealth of information to consider is a double-edged-sword: many properties are considered that may not be useful for determining similarity, adding distractions that can decrease performance. The Abstract method has an additional advantage over the graph embeddings in that it is less restricted in terms of the kind of information it can consider, whereas the graph embeddings focus solely on relations and can not make use of literals directly.

Our topological models (OT category) perform better than the KGE and worse than the LME models on the WD-WordSim353 dataset, and the worst among the categories on the other two datasets. Class performs better than Jiang Conrath on two out of three datasets, though the difference between the methods is not significant. Generally speaking, TopSim is able to combine the different regions in a way that outperforms most of the individual models, and it clearly outperforms the methods that only rely on ontological structure (Class and Jiang Conrath). However, TopSim consistently performs worse than the composite embeddings. Among the combinations, Composite-Best performs the best, indicating that combining a small set of reliable models may be a better strategy than composing a larger set of embeddings together. However, here we note that the composite embeddings do not improve over the Abstract LME score, despite the fact that they include Abstract as one of their embeddings. This indicates that it is difficult to combine models that consider additional information without adding noise that decreases the utility.

Table 4

Impact of retrofitting across the different benchmarks. Here we show results on retrofitting with *WD-all*, where the edges are scored with BERT-based cosine similarity. Highest Kendall-Tau (KT) values and increases per column are marked in bold

Type	Model	WD-WordSim353			WD-RG65			WD-MC30			Avg Δ
		Old KT	New KT	Δ	Old KT	New KT	Δ	Old KT	New KT	Δ	
KGE	TransE	0.220	0.212	−0.008	0.182	0.132	−0.05	0.133	0.167	0.033	−0.008
	ComplEx	0.208	0.237	0.029	0.161	0.193	0.032	0.250	0.233	−0.017	0.015
	Deepwalk	0.281	0.323	0.042	0.238	0.212	−0.026	0.291	0.291	0	0.005
	S-Deepwalk	0.042	0.099	0.058	0.030	0.124	**0.093**	−0.143	0.067	**0.210**	**0.120**
LME	Abstract	0.523	**0.567**	0.044	**0.518**	**0.497**	−0.021	**0.567**	**0.583**	0.017	0.013
	Lexicalization	0.374	0.381	0.007	0.408	0.397	−0.011	0.450	0.400	−0.050	−0.018
	Label	0.041	0.140	**0.099**	−0.032	0.029	0.061	0.167	0.267	0.100	0.087
	Label + Desc	0.368	0.448	0.080	0.132	0.154	0.021	0.25	0.283	0.033	0.045
Comb.	Composite-All	0.437	0.489	0.052	0.304	0.275	−0.029	0.250	0.317	0.067	0.030
	Composite-Best	**0.483**	0.533	0.050	0.393	0.411	0.018	0.483	0.517	0.033	0.034

5.2. *What is the impact of retrofitting?*

Retrofitting is overall beneficial for estimating similarity (Table 4). On average across the three benchmarks, it improves the performance of nine out of the eleven methods. The highest overall improvement is observed for the S-Deepwalk method, whose Kendall-Tau score on the WD-MC30 benchmark is increased by 0.2. Despite this bump, the new S-DeepWalk score is still relatively low (0.067). We also note a consistent improvement with the simpler methods, like Label and Label + Desc, which can be expected given that these methods do not consider taxonomic information sufficiently before retrofitting. For example, the distance between dissimilar objects, like credit and card, is nearly the same before and after retrofitting the Label method, whereas the distance between highly similar objects like money and cash decreases significantly (from 3.7 to 2.2, on a scale where 4 is the maximum and 1 is the minimum). The impact of retrofitting is lower on methods that consider richer information already, like Abstract and Lexicalized. This is because these methods already integrate taxonomic information, and retrofitting might bring concepts that are nearly identical or merely related too close in the embedding space. For instance, retrofitting decreases the distance between seafood and lobster from 2.8 to 1.3. Still, the impact of retrofitting on Abstract is positive on two out of three benchmarks, leading to the new top results on the benchmarks WD-WordSim353 and WD-MC30.

5.3. *What knowledge is most beneficial for retrofitting?*

We analyze the impact of different retrofitting knowledge sources in Table 5. Among the Wikidata variants, we observe that retrofitting with child-parent data performs comparable to using both child-parent and sibling data together, and this finding is consistent across the methods. This result indicates that WD-sibling data is less useful for retrofitting of models compared to parent-child data. We believe that this observation is due to the Wisdom-of-the-crowd [86] knowledge creation method of Wikidata, which results in a wide ontology with many children per parent. Each of the child-parent relations is reasonable and connects two relatively similar concepts, but two children connecting to the same parent may be dissimilar and may specialize the parent node in different dimensions [39]. We illustrate this reasoning with the following example. Fairy tale (Q699) is a child of the concept tale (Q17991521), and the similarity between the two concepts is relatively high. At the same time, other children of tale include old-fashioned tale, cumulative tale, urban tale, and Zeichengeschichte (a German television genre), whose similarity with fairy tale is lower, as each of these siblings describes a different aspect of a tale: its location, ethnic tradition, form, or genre. To quantify this phenomenon, we show similarity scores with the KGTK similarity GUI [40] between fairy tale, and its parent (tale) and siblings in Fig. 4. As apparent in the figure, the similarity between the child and the parent (fairy tale and tale) is clearly higher than any of the similarities between two siblings (e.g., fairy tale and old-fashioned tale), according to any of the similarity metrics.

Table 5
Impact of different retrofitting knowledge variants on the WD-WordSim353 dataset. Highest Kendall-Tau (KT) increases per column are marked in bold

Type	Model	/	WD-all		WD-child-parent		WD-siblings		ProBase	
		KT	KT	Δ	KT	Δ	KT	Δ	KT	Δ
KGE	TransE	0.220	0.212	−0.008	0.210	−0.010	0.212	−0.008	0.092	−0.128
	ComplEx	0.208	0.237	0.029	0.248	0.040	0.219	0.011	0.123	−0.085
	Deepwalk	0.281	0.323	0.042	0.323	0.042	0.296	0.015	0.220	−0.061
	S-Deepwalk	0.042	0.099	0.057	0.125	0.083	0.04	−0.002	−0.091	−0.133
LME	Abstract	**0.523**	**0.567**	0.044	**0.569**	0.046	**0.521**	−0.002	0.462	−0.061
	Lexicalization	0.374	0.381	0.007	0.372	−0.002	0.373	−0.001	0.273	−0.101
	Label	0.041	0.140	**0.099**	0.156	**0.115**	0.07	0.029	0.037	**−0.004**
	Label + Desc	0.368	0.448	0.080	0.457	0.089	0.401	0.033	0.318	−0.050
Comb.	Composite-6	0.437	0.489	0.052	0.488	0.051	0.471	**0.034**	0.369	−0.068
	Composite-2	0.483	0.533	0.050	0.529	0.046	0.500	0.017	**0.464**	−0.019

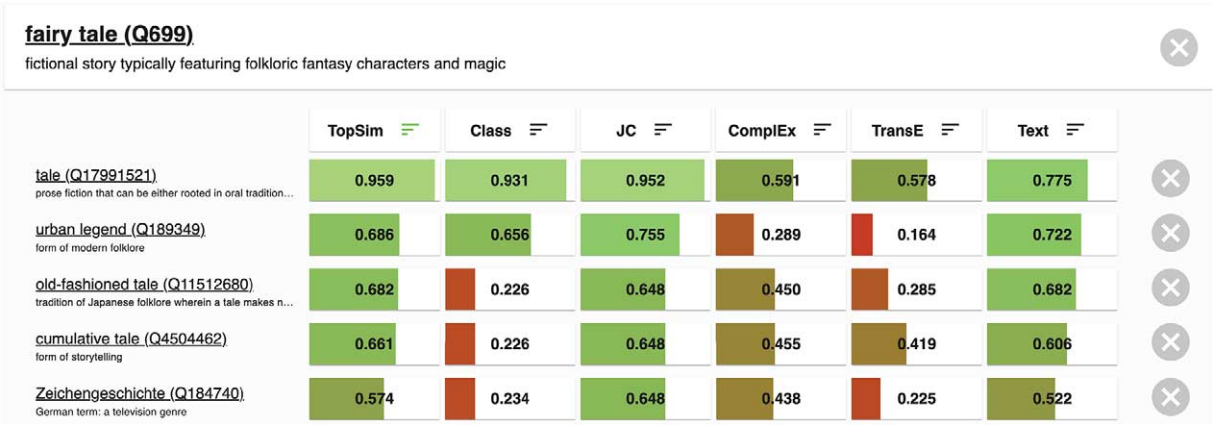

Fig. 4. Similarity scores with the KGTK similarity GUI [40] between fairy tale, and its parent (tale) and siblings.

Retrofitting with ProBase's IsA relations yields consistently negative results across all methods. This could be due to the quality of the underlying data, our choice to use the relation counts as similarity estimates, or the imperfect mapping of ProBase nodes to Wikidata. Comparing the results across the different methods, we again observe that the simpler methods and the composite methods benefit most from retrofitting, whereas the more elaborate methods benefit from retrofitting much less.

6. Discussion and future work

In summary, our evaluation of a variety of KGE, LME, OT, and combinations of models reveals several key insights. First, language models are strong indicators of concept similarity in KGs, however, they are extremely sensitive to the kind of input that they operate on. Therefore, scalable and reliable lexicalization is a key component of LME-based similarity models. Second, KGE models, which largely transfer knowledge about instances to concepts, are also strong indicators of similarity, but not encoding the literal content is a key limitation for these models. Third, retrofitting is helpful across the board, though its impact is larger for simpler models that originally did not encode structural information. Fourth, we note that careful selection of knowledge for retrofitting is essential, given the size of the sources of background knowledge and their creation methods. Here, we note that parent-child relations from Wikidata are most useful for retrofitting, whereas knowledge from ProBase generally hurts performance.

Looking forward, we discuss four key considerations for reliable methods and evaluation of concept similarity over Wikidata: dimensional metrics of similarity, extending coverage to entity nodes, large-scale evaluation, and downstream applications.

From a single number to a set of dimensions Word and concept similarity tasks have typically strived to produce a single similarity score that aggregates over all aspects of two concepts. However, concepts may be similar in different ways, i.e., they may have similar parts, utility, physical appearance, typical location, and so on. Moreover, concepts may be antonyms or opposites, which may not map easily to a single numeric similarity scale. Applications that reason over concepts, e.g., for search or intelligent analytics, can benefit from a dimensional notion of similarity that scores concepts with an interpretable similarity vector rather than a single scalar. An opening for this idea is provided by prior work on dimensions of commonsense knowledge [38] and concept norms benchmarks with human judgments about salient concept features [19].

From concepts to all Wikidata nodes In this work, we focus on concepts in Wikidata, and leave out entities from the current evaluation. A comprehensive similarity framework should include entities as well, especially given that these are prevalent in Wikidata. While our metrics and models technically work for entities too, it is an empirical question whether they will generalize well, given that concepts and instances are conceptually different despite certain overlaps [74,82]. One way to qualify these differences is by following up on our first point and devising a set of dimensions that qualify entities as well. Here, we believe that the relations found in Wikidata provide a useful entry point for creating a vector of entity dimensions, though certain engineering may be needed to make sure that the dimensions are meaningful and complete.

From small-scale to representative evaluation The most urgent obstacle to a representative evaluation of concept similarity is the lack of large-scale evaluation data. Creating evaluation data for similarity is extremely laborious, and typically entails a pairwise comparison of conceptual pairs, which is difficult to perform at scale. It is unclear how to create a large-scale evaluation dataset for Wikidata, which would ideally also include multiple dimensions of similarity (point 1) and both concepts and entities (point 2). We attempted to infer data for evaluating similarity from sources of commonsense knowledge, like ConceptNet [84], but this appeared to be difficult in practice due to the excessive noise present in these sources. The CSLB property norms dataset [19] provides a positive example of how similarity judgments can be elicited from humans, but it still remains to map these judgments to Wikidata, and their size is still relatively small. Using the data-to-text idea [93], one can leverage the Wikidata structure to generate a large number of pairs, however, it is not clear how to generate similarity judgments, given the inconsistent modeling of Wikidata [39].

From similarity tasks to applications Besides the need to test the generalizability of our findings on a broader set of Wikidata nodes (entities) and on larger evaluation sets, we suggest the integration of explicit similarity metrics for downstream applications. Robust metrics of concept and entity similarity are essential to improve the quality of knowledge graphs. Concept similarity metrics can be applied retrospectively to detect existing duplicates in large knowledge graphs, as well as to provide similar suggestions to human or bot editors when entering new information [82,103]. Concept similarity can be instrumental in providing alignment and recommendations for a variety of application domains, including food, biomedicine, and science. Namely, concept similarity can be used to customize or personalize a recipe for a user, by replacing an ingredient with a similar one that is preferred or more accessible to the user [18]. In the domain of biomedicine, concept similarity can be applied to connect pieces of knowledge between different modalities of data, including documents and domain ontologies [22]. In science, concept similarity can be used to perform fair automatic screening and matching between reviewers and papers based on the similarity of the reviewer's work [43]. While devising effective methods for these tasks will likely require substantial engineering, we expect that the framework and the insights provided in our paper will be beneficial for those efforts.

7. Conclusions

This paper designed a framework with representative models for estimating the similarity of concepts in Wikidata. We considered language model embeddings, knowledge graph embeddings, and topological information. We

developed combinations of these models, and experimented with knowledge injection via retrofitting to two large knowledge graphs: Wikidata and ProBase. The experiments revealed that pairing language models with well-curated information found in abstracts led to optimal performance. Balancing between information wealth and noise, on the one hand, and between structure and content, on the other hand, are important considerations for future model engineering. We found that retrofitting with taxonomic information from Wikidata generally improved performance across methods, with simpler methods benefiting more from retrofitting. Retrofitting with the ProBase KG yielded consistently negative results, indicating that the impact of retrofitting directly depends on the quality of the underlying data. Future work should investigate contextual similarity between concepts, which would characterize the partial identity of concept and entity pairs. The key obstacle to developing reliable metrics for concept similarity in Wikidata lies in the lack of representative evaluation – addressing this challenge is a high-priority task for subsequent research. Finally, applying the similarity metrics at scale for downstream reasoning tasks is a necessary follow-up step to understand their potential for impact and their latent shortcomings.

References

[1] O. Agarwal, H. Ge, S. Shakeri and R. Al-Rfou, Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training, 2020, arXiv preprint arXiv:2010.12688.

[2] E. Agirre and A. Soroa, Personalizing pagerank for word sense disambiguation, in: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 2009, pp. 33–41, .

[3] H. Al-Mubaid and H.A. Nguyen, A cluster-based approach for semantic similarity in the biomedical domain, in: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2006, pp. 2713–2717. doi:10.1109/IEMBS.2006.259235.

[4] M.A. Alkhamees, M.A. Alnuem, S.M. Al-Saleem and A.M. Al-Ssulami, A semantic metric for concepts similarity in knowledge graphs, *Journal of Information Science* (2021). 01655515211020580.

[5] C.F. Baker, C.J. Fillmore and J.B. Lowe, The Berkeley framenet project, in: *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998.

[6] M. Baroni and A. Lenci, Distributional memory: A general framework for corpus-based semantics, *Computational Linguistics* **36**(4) (2010), 673–721. doi:10.1162/coli_a_00016.

[7] D. Bollegala, Y. Matsuo and M. Ishizuka, Measuring semantic similarity between words using web search engines, in: *WWW 07*, 2007, pp. 757–766.

[8] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, Translating embeddings for modeling multi-relational data, *Advances in neural information processing systems* **26** (2013).

[9] A. Budanitsky and G. Hirst, Evaluating wordnet-based measures of lexical semantic relatedness, *Computational linguistics* **32**(1) (2006), 13–47. doi:10.1162/coli.2006.32.1.13.

[10] M. Caballero and A. Hogan, Global vertex similarity for large-scale knowledge graphs, in: *Wikidata@ ISWC*, 2020.

[11] J. Cheng, Z. Wang, J.-R. Wen, J. Yan and Z. Chen, Contextual text understanding in distributional semantic space, in: *ACM International Conference on Information and Knowledge Management (CIKM)*, ACM – Association for Computing Machinery, 2015, https://www.microsoft.com/en-us/research/publication/contextual-text-understanding-in-distributional-semantic-space/.

[12] N. Cheniki, A. Belkhir, Y. Sam and N. Messai, Lods: A linked open data based similarity measure, in: *2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, IEEE, 2016, pp. 229–234.

[13] M.T. Chi, P.J. Feltovich and R. Glaser, Categorization and representation of physics problems by experts and novices, *Cognitive science* **5**(2) (1981), 121–152. doi:10.1207/s15516709cog0502_2.

[14] K. Church and P. Hanks, Word association norms, mutual information, and lexicography, *Computational linguistics* **16**(1) (1990), 22–29.

[15] M. Čuljak, A. Spitz, R. West and A. Arora, Strong heuristics for named entity linking, in: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 2022, pp. 235–246. https://aclanthology.org/2022.naacl-srw.30. doi:10.18653/v1/2022.naacl-srw.30.

[16] Y. Dai, S. Wang, N.N. Xiong and W. Guo, A survey on knowledge graph embedding: Approaches, applications and benchmarks, *Electronics* **9**(5) (2020), 750. doi:10.3390/electronics9050750.

[17] C. d'Amato, S. Staab and N. Fanizzi, On the influence of description logics ontologies on conceptual similarity, in: *International Conference on Knowledge Engineering and Knowledge Management*, Springer, 2008, pp. 48–63.

[18] J. DeMiguel, L. Plaza and B. Díaz-Agudo, ColibriCook: A CBR system for ontology-based recipe retrieval and adaptation, in: *ECCBR Workshops*, 2008, https://api.semanticscholar.org/CorpusID:6441817.

[19] B.J. Devereux, L.K. Tyler, J. Geertzen and B. Randall, The Centre for Speech, Language and the Brain (CSLB) concept property norms, *Behavior research methods* **46**(4) (2014), 1119–1127. doi:10.3758/s13428-013-0420-4.

[20] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[21] M. Faruqui, J. Dodge, S.K. Jauhar, C. Dyer, E. Hovy and N.A. Smith, *Retrofitting Word Vectors to Semantic Lexicons*, 2015.

[22] H. Fei, Y. Ren, Y. Zhang, D. Ji and X. Liang, Enriching contextualized language model from knowledge graph for biomedical information extraction, *Briefings in bioinformatics* **22**(3) (2021), bbaa110. doi:10.1093/bib/bbaa110.
[23] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman and E. Ruppin, Placing search in context: The concept revisited, *ACM Transactions on Information Systems* **20**(1) (2002), 116–131. doi:10.1145/503104.503110.
[24] J.L. Fleiss, Measuring nominal scale agreement among many raters, *Psychological bulletin* **76**(5) (1971), 378. doi:10.1037/h0031619.
[25] E. Gabrilovich, S. Markovitch et al., Computing semantic relatedness using Wikipedia-based explicit semantic analysis, in: *IJcAI*, Vol. 7, 2007, pp. 1606–1611.
[26] M. Galkin, P. Trivedi, G. Maheshwari, R. Usbeck and J. Lehmann, Message passing for hyper-relational knowledge graphs, in: *EMNLP*, 2020.
[27] D. Gentner, Structure-mapping: A theoretical framework for analogy, *Cognitive science* **7**(2) (1983), 155–170.
[28] D. Gentner, Metaphor as structure mapping: The relational shift, *Child development* (1988), 47–59. doi:10.2307/1130388.
[29] D. Gentner and A.B. Markman, Structure mapping in analogy and similarity, *American psychologist* **52**(1) (1997), 45. doi:10.1037/0003-066X.52.1.45.
[30] J. Goikoetxea, E. Agirre and A. Soroa, Single or multiple? Combining word representations independently learned from text and wordnet, in: *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[31] J. Goikoetxea, A. Soroa and E. Agirre, Random walks and neural network language models on knowledge bases, in: *Proceedings of the 2015*, 2015, pp. 1434–1439, conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies.
[32] N. Goodman, *Seven Strictures on Similarity*, 1972.
[33] A. Grover and J. Leskovec, node2vec: Scalable feature learning for networks, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 855–864. doi:10.1145/2939672.2939754.
[34] K. Han, T.C. Ferreira and C. Gardent, Generating questions from Wikidata triples, in: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2022, pp. 277–290.
[35] S. Harispe, S. Ranwez, S. Janaqi and J. Montmain, *Semantic Similarity from Natural Language and Ontology Analysis*, Synthesis Lectures on Human Language Technologies 2015, pp. 1–254. http://arxiv.org/abs/1704.05295. arXiv:1704.05295. doi:10.1007/978-3-031-02156-5.
[36] F. Ilievski, D. Garijo, H. Chalupsky, N.T. Divvala, Y. Yao, C. Rogers, R. Li, J. Liu, A. Singh, D. Schwabe and P. Szekely, KGTK: A toolkit for large knowledge graph manipulation and analysis, in: *International Semantic Web Conference*, Springer, Cham, 2020, pp. 278–293.
[37] F. Ilievski, E. Hovy, P. Vossen, S. Schlobach and Q. Xie, The role of knowledge in determining identity of long-tail entities, *Journal of Web Semantics* **61** (2020), 100565. doi:10.1016/j.websem.2020.100565.
[38] F. Ilievski, A. Oltramari, K. Ma, B. Zhang, D.L. McGuinness and P. Szekely, Dimensions of commonsense knowledge, *Knowledge-Based Systems* **229** (2021), 107347. doi:10.1016/j.knosys.2021.107347.
[39] F. Ilievski, J. Pujara and K. Shenoy, Does Wikidata Support Analogical Reasoning? in: *KGSWC*, 2022.
[40] F. Ilievski, P. Szekely, G. Satyukov and A. Singh, User-friendly comparison of similarity algorithms on Wikidata, 2021, arXiv preprint arXiv:2108.05410.
[41] F. Ilievski, P. Vossen and S. Schlobach, Systematic study of long tail phenomena in entity linking, in: *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 664–674.
[42] N. Jain, J.-C. Kalo, W.-T. Balke and R. Krestel, Do embeddings actually capture knowledge graph semantics? in: *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*, Springer, 2021, pp. 143–159. doi:10.1007/978-3-030-77385-4_9.
[43] X. Ji, A. Ritter and P.-Y. Yen, Using ontology-based semantic similarity to facilitate the article screening process for systematic reviews, *Journal of biomedical informatics* **69** (2017), 33–42. doi:10.1016/j.jbi.2017.03.007.
[44] J.J. Jiang and D. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in: *ROCLING/IJCLCLP*, 1997.
[45] N. Klein, F. Ilievski and P. Szekely, Generating explainable abstractions for Wikidata entities, in: *Proceedings of the 11th on Knowledge Capture Conference*, 2021, pp. 89–96. doi:10.1145/3460210.3493580.
[46] A. Kristiadi, M.A. Khan, D. Lukovnikov, J. Lehmann and A. Fischer, Incorporating literals into knowledge graph embeddings, in: *The Semantic Web – ISWC 2019: Proceedings, Part I 18*, 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Springer, 2019, pp. 347–363.
[47] T.K. Landauer, P.W. Foltz and D. Laham, An introduction to latent semantic analysis, *Discourse processes* **25**(2–3) (1998), 259–284.
[48] J.J. Lastra-Díaz, J. Goikoetxea, M.A.H. Taieb, A. García-Serrano, M.B. Aouicha and E. Agirre, A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art, *Engineering Applications of Artificial Intelligence* **85** (2019), 645–665. doi:10.1016/j.engappai.2019.07.010.
[49] C. Leacock and M. Chodorow, Combining local context and WordNet similarity for word sense identification, *WordNet: An electronic lexical database* **49**(2) (1998), 265–283.
[50] P. Lee, L.V. Lakshmanan and J.X. Yu, On top-k structural similarity search, in: *2012 IEEE 28th International Conference on Data Engineering*, IEEE, 2012, pp. 774–785. doi:10.1109/ICDE.2012.109.
[51] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint, 2019, arXiv:1907.11692.
[52] K. Lund and C. Burgess, Producing high-dimensional semantic spaces from lexical co-occurrence, *Behavior research methods, instruments, & computers* **28**(2) (1996), 203–208. doi:10.3758/BF03204766.
[53] A. Maedche and S. Staab, Ontology learning for the semantic web, *IEEE Intelligent systems* **16**(2) (2001), 72–79. doi:10.1109/5254.920602.

[54] D.L. Medin, R.L. Goldstone and D. Gentner, Respects for similarity, *Psychological review* **100**(2) (1993), 254. doi:10.1037/0033-295X.100.2.254.
[55] R. Meymandpour and J.G. Davis, Enhancing recommender systems using linked open data-based semantic analysis of items, in: *AWC*, 2015, pp. 11–17.
[56] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, in: *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.
[57] G.A. Miller, WordNet: A lexical database for English, *Communications of the ACM* **38**(11) (1995), 39–41. doi:10.1145/219717.219748.
[58] G.A. Miller and W.G. Charles, Contextual correlates of semantic similarity, *Language and cognitive processes* **6**(1) (1991), 1–28. doi:10.1080/01690969108406936.
[59] N. Mrkšić, I. Vulić, D.O. Séaghdha, I. Leviant, R. Reichart, M. Gašić, A. Korhonen and S. Young, Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints, *Transactions of the association for Computational Linguistics* **5** (2017), 309–324. doi:10.1162/tacl_a_00063.
[60] G.L. Murphy and D.L. Medin, The role of theories in conceptual coherence, *Psychological review* **92**(3) (1985), 289. doi:10.1037/0033-295X.92.3.289.
[61] S. Natarajan, S. Vairavasundaram, S. Natarajan and A.H. Gandomi, Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data, *Expert Systems with Applications* **149** (2020), 113248. doi:10.1016/j.eswa.2020.113248.
[62] C.E. Osgood, The similarity paradox in human learning: A resolution, *Psychological review* **56**(3) (1949), 132. doi:10.1037/h0057488.
[63] A. Passant, Measuring semantic distance on linking data and using it for resources recommendations, in: *2010 AAAI Spring Symposium Series*, 2010.
[64] A. Passant, dbrec – music recommendations using DBpedia, in: *International Semantic Web Conference*, Springer, 2010, pp. 209–224.
[65] T. Pedersen, S.V. Pakhomov, S. Patwardhan and C.G. Chute, Measures of semantic similarity and relatedness in the biomedical domain, *Journal of biomedical informatics* **40**(3) (2007), 288–299. doi:10.1016/j.jbi.2006.06.004.
[66] V. Pekar and S. Staab, Taxonomy learning-factoring the structure of a taxonomy into a semantic classification decision, in: *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.
[67] J. Pennington, R. Socher and C.D. Manning, Glove: Global vectors for word representation, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. doi:10.3115/v1/D14-1162.
[68] B. Perozzi, R. Al-Rfou and S. Skiena, DeepWalk: Online learning of social representations, 2014, arXiv:1403.6652.
[69] M.E. Peters, M. Neumann, R.L. Logan IV., R. Schwartz, V. Joshi, S. Singh and N.A. Smith, Knowledge enhanced contextual word representations, 2019, arXiv preprint arXiv:1909.04164.
[70] A. Petrova, E.V. Kostylev, B.C. Grau and I. Horrocks, Towards explainable entity matching via comparison queries, in: *OM@ ISWC*, 2019, pp. 197–198.
[71] A. Petrova, E.V. Kostylev, B.C. Grau and I. Horrocks, Query-based entity comparison in knowledge graphs revisited, in: *International Semantic Web Conference*, Springer, 2019, pp. 558–575.
[72] G. Pirró, Reword: Semantic relatedness in the web of data, in: *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
[73] G. Pirró and N. Seco, Design, implementation and evaluation of a new semantic similarity metric combining features and intrinsic information content, in: *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, Springer, 2008, pp. 1271–1288.
[74] A. Piscopo and E. Simperl, Who models the world? Collaborative ontology creation and user roles in Wikidata, *Proceedings of the ACM on Human-Computer Interaction* **2**(CSCW) (2018), 1–18.
[75] R. Rada, H. Mili, E. Bicknell and M. Blettner, Development and application of a metric on semantic nets, *IEEE transactions on systems, man, and cybernetics* **19**(1) (1989), 17–30. doi:10.1109/21.24528.
[76] N. Reimers and I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics*, 2019, https://arxiv.org/abs/1908.10084.
[77] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, 1995, arXiv preprint cmp-lg/9511007.
[78] D.L.T. Rohde, L.M. Gonnerman and D.C. Plaut, An improved model of semantic similarity based on lexical co-occurence, *Communications of the Acm* **8** (2006), 627–633.
[79] E.M. Roth and E.J. Shoben, The effect of context on the structure of categories, *Cognitive psychology* **15**(3) (1983), 346–378. doi:10.1016/0010-0285(83)90012-9.
[80] H. Rubenstein and J.B. Goodenough, Contextual correlates of synonymy, *Communications of the ACM* **8**(10) (1965), 627–633. doi:10.1145/365628.365657.
[81] J. Shen, C. Wang, L. Gong and D. Song, Joint language semantic and structure embedding for knowledge graph completion, 2022, arXiv preprint arXiv:2209.08721.
[82] K. Shenoy, F. Ilievski, D. Garijo, D. Schwabe and P. Szekely, A study of the quality of Wikidata, *Journal of Web Semantics* **72** (2022), 100679. doi:10.1016/j.websem.2021.100679.
[83] L.B. Smith and D. Heise, Perceptual similarity and conceptual structure, in: *Advances in Psychology*, Vol. 93, Elsevier, 1992, pp. 233–272.
[84] R. Speer, J. Chin and C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
[85] Z. Sun, Z.-H. Deng, J.-Y. Nie and J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, 2019, arXiv preprint arXiv:1902.10197.

[86] J. Surowiecki, *The Wisdom of Crowds: Why the Many Are Smarter than the Few and How Collective Wisdom Shapes Business, Economies, Societies, and Nations*, Doubleday, New York, 2004. ISBN 9780385503860.
[87] A. Thawani, M. Hu, E. Hu, H. Zafar, N.T. Divvala, A. Singh, E. Qasemi, P.A. Szekely and J. Pujara, Entity linking to knowledge graphs to infer column types and properties, *SemTab@ ISWC* **2019** (2019), 25–32.
[88] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier and G. Bouchard, Complex embeddings for simple link prediction, in: *International Conference on Machine Learning*, PMLR, 2016, pp. 2071–2080.
[89] L. Tu, G. Lalwani, S. Gella and H. He, An empirical study on robustness to spurious correlations using pre-trained language models, *Transactions of the Association for Computational Linguistics* **8** (2020), 621–633. doi:10.1162/tacl_a_00335.
[90] P.D. Turney, Similarity of semantic relations, *Computational Linguistics* **32**(3) (2006), 379–416. doi:10.1162/coli.2006.32.3.379.
[91] A. Tversky, Features of similarity, *Psychological review* **84**(4) (1977), 327. doi:10.1037/0033-295X.84.4.327.
[92] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, *Advances in neural information processing systems* **30** (2017).
[93] P. Vossen, F. Ilievski, M. Postma and R. Segers, Don't annotate, but validate: A data-to-text method for capturing event data, in: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
[94] D. Vrandečić and M. Krötzsch, Wikidata: A free collaborative knowledgebase, *Communications of the ACM* **57**(10) (2014), 78–85. doi:10.1145/2629489.
[95] J. Wang, F. Ilievski, P. Szekely and K.-T. Yao, Augmenting knowledge graphs for better link prediction, *IJCAI* (2022).
[96] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge graph embedding: A survey of approaches and applications, *IEEE Transactions on Knowledge and Data Engineering* **29**(12) (2017), 2724–2743. doi:10.1109/TKDE.2017.2754499.
[97] R. Wang, H.K. Cheng, Y. Jiang and J. Lou, A novel matrix factorization model for recommendation with LOD-based semantic similarity measure, *Expert Systems with Applications* **123** (2019), 70–81. doi:10.1016/j.eswa.2019.01.036.
[98] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge graph and text jointly embedding, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1591–1601. doi:10.3115/v1/D14-1167.
[99] C. Wei, B. Wu, S. Wang, R. Lou, C. Zhan, F. Li and Y. Cai, Analyticdb-v: A hybrid analytical engine towards query fusion for structured and unstructured data, *Proceedings of the VLDB Endowment* **13**(12) (2020), 3152–3165. doi:10.14778/3415478.3415541.
[100] Z. Wu and M. Palmer, Verb semantics and lexical selection, 1994, arXiv preprint arXiv:cmp-lg/9406033.
[101] G. Xu, Q. Zhang, D. Yu, S. Lu and Y. Lu, JKRL: Joint knowledge representation learning of text description and knowledge graph, *Symmetry* **15**(5) (2023), 1056. doi:10.3390/sym15051056.
[102] L. Yao, C. Mao and Y. Luo, KG-BERT: BERT for knowledge graph completion, 2019, arXiv preprint arXiv:1909.03193.
[103] B. Zhang, F. Ilievski and P. Szekely, Enriching Wikidata with linked open data, 2022, in: Wikidata-22 workshop.
[104] H. Zhang, T. Zheng, J. Gao, C. Miao, L. Su, Y. Li and K. Ren, Data poisoning attack against knowledge graph embedding, 2019, arXiv preprint arXiv:1904.12052.
[105] X. Zhang, Q. Yang, J. Ding and Z. Wang, Entity profiling in knowledge graphs, *IEEE Access* **8** (2020), 27257–27266. doi:10.1109/ACCESS.2020.2971567.
[106] Z. Zhou, C. Wang, Y. Feng and D. Chen, JointE: Jointly utilizing 1D and 2D convolution for knowledge graph embedding, *Knowledge-Based Systems* **240** (2022), 108100. doi:10.1016/j.knosys.2021.108100.
[107] G. Zhu and C.A. Iglesias, Computing semantic similarity of concepts in knowledge graphs, *IEEE Transactions on Knowledge and Data Engineering* **29**(1) (2016), 72–85. doi:10.1109/TKDE.2016.2610428.