Efficient Graph-based Document Similarity

Christian Paul[1], Achim Rettinger[1], Aditya Mogadala[1],
Craig Knoblock[2], Pedro Szekely[2]

[1] Institute of Applied Informatics and Formal Description Methods (AIFB),
Karlsruhe Institute for Technology, 76131 Karlsruhe, Germany
[2] Information Sciences Institute, University of Southern California, Marina Del Rey,
CA 90292

Abstract. Assessing the relatedness of documents is at the core of many applications such as document retrieval and recommendation. Most similarity approaches operate on word-distribution based document representations - fast to compute, but problematic when documents differ in language, vocabulary or type and neglecting the rich relational knowledge available in Knowledge Graphs. In contrast, graph-based document models can leverage valuable knowledge about relations between entities - however, due to expensive graph operations, similarity assessments tend to become infeasible in many applications. This paper presents an efficient semantic similarity approach exploiting explicit hierarchical and transversal relations. We show in our experiments that (i) our similarity measure provides a significantly higher correlation with human notions of document similarity than comparable measures, (ii) this also holds for short documents with few annotations, (iii) document similarity can be calculated efficiently compared to other graph-traversal based approaches.

Keywords: Semantic Document Similarity, Knowledge Graph based Document Models, Efficient Similarity Calculation

1 Introduction

Searching for related documents given a query document is a common task for applications in many domains. For example, a news website might want to recommend content with regards to the article a user is reading. Implementing such functionality requires (i) an efficient method to locate relevant documents out of a possibly large corpus and (ii) a notion of document similarity. Established approaches measure text similarity statistically, based on the distributional hypothesis , which states that words occurring in the same context tend to be similar in meaning. By inferring semantics from text without using explicit knowledge, word-level approaches become susceptible to problems caused by polysemy (ambiguous terms) and synonymy (words with similar meaning)[22].

Another problem arises when using distributional measures across heterogeneous documents: due to different vocabularies and text length (e.g. news

articles, Tweets) or languages, each type may underlie a different word distribution, making them hard to compare. Also, documents of different modalities (images, video, audio) may provide metadata, but no continuous text at all.

Semantic technologies help to address both these shortcomings. Knowledge bases like DBPedia[3] or Wikidata[4] unambiguously describe millions of entities and their relationship as a semantic graph. Using tools such as the cross-lingual text annotator xLisa[23], documents of different natures can be represented in the common format of knowledge graph entities. By using entities instead of text, heterogeneous content can be handled in an integrated manner and some disadvantages of statistical similarity approaches can be avoided.

In this paper, we present a scalable approach for related-document search using entity-based document similarity. In a pre-processing step called *Semantic Document Expansion*, we enrich annotated documents with hierarchical and transversal relational knowledge from a knowledge graph (Sec. 2). At search time, we retrieve a candidate set of semantically expanded documents using an inverted index (Sec. 3). Based on the added semantic knowledge, we find paths between annotations and compute path-based semantic similarity (Sec. 3.1, 3.2). By performing graph traversal steps only during pre-processing, we overcome previous graph-based approaches' performance limitations.

We evaluate the performance of our document similarity measure on two different type of data sets. First, we show on the standard benchmark for document-level semantic similarity that our knowledge-based similarity method significantly outperforms all related approaches (sec. 5.2). Second, we demonstrate that we even achieve superior performance on sentence-level semantic similarity, as long as we find at least one entity to represent the sentence (Sec. 5.3). This suggest, that with growing knowledge graphs and improving entity linking tools, document models based on explicit semantics become competitive compared to the predominant vector-space document models based on implicit semantics.

2 Knowledge Graph based Document Model

Given an document annotated with knowledge graph entities, Semantic Document Expansion enriches the annotations with relational knowledge. Following Damljanovic et al.[8], we distinguish between two types of exploited knowledge depending on the type of edge that is traversed to obtain it: an edge is classified as **hierarchical** if it represents a child-parent-relationship and denotes membership of an entity in a class or category. A **transversal** edge expresses a semantic, non-hierarchical predicate. Both groups of edge types have the potential to add value to our semantic measures: whereas connectivity via hierarchical edges indicates common characteristics on some categorical level, transversal paths express a relationship between entities independent of their intrinsic or type-based relatedness.

[3] http://wiki.dbpedia.org/Datasets2014
[4] https://www.wikidata.org

2.1 Hierarchical expansion

Hierarchically expanding an entity means enriching it with all information required for hierarchical similarity computation, so that it can be performed between any two expanded entities without accessing the knowledge graph. For each of a document's annotations, we locate its position within the hierarchical subgraph of our knowledge base and add all its *parent* and *ancestor* elements to it. Figure 1 shows a hierarchically expanded DBPedia entity using the Wikipedia Category System, with its parents and ancestors signaled by rectangular nodes.

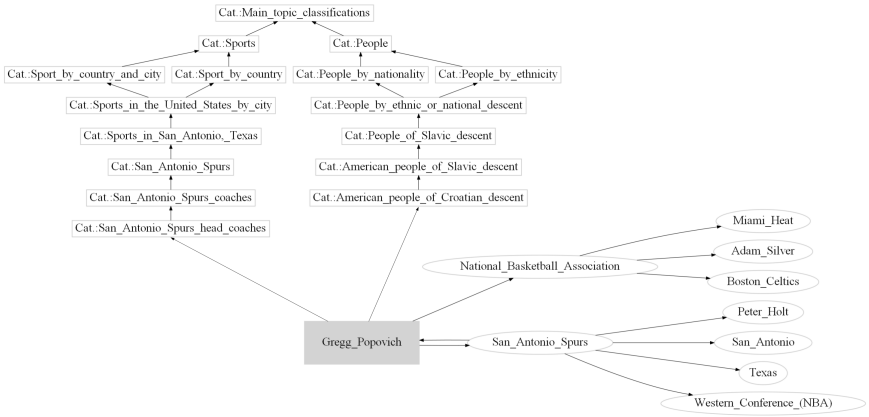


Fig. 1. Excerpt of hierarchically (top left) and transversally (bottom right, expansion radius=2) expanded DBPedia entity `Gregg_Popovich`. (Cat.= Category)

2.2 Transversal expansion

Transversal expansion resembles the Spreading Activation method: starting from a knowledge graph entity, it traverses semantic, non-hierarchical edges for a fixed number L of steps, while weighting and adding encountered entities to the document. We call L the entity's **expansion radius**. Formally, for each document annotation a, for each entity e encountered in the process, a weight is assigned according to the formula $w_a(e) = \sum_{l=1}^{L} \beta^l * |paths_{a,e}^{(l)}|$. Paths of length l are penalized by a factor of β^l, expressing the intuition that more distant entities are less relevant to annotation a than closer ones. Also, the more paths connect a with e, the higher a weight is assigned. We consider only outgoing edges - although this risks missing connections, we argue it also significantly reduces noise since the indegree for knowledge graph nodes can be very high (e.g. DBPedia entity `United_States`: ≈220k). This weighting notion is closely related to the graph measure *Katz centrality* [12]. Nunes et al. used the same principle to determine connectivity between two nodes[14]. We set $\beta = 0.5$, as this value yielded the best results in our experiments.

3 Semantic Document Similarity

Our main contribution is an approach for efficient assessment of document relatedness based on entity-level semantic similarity. In order to provide a scalable solution, it is essential that at search time, only light-weight tasks are performed. Assuming we have a corpus of semantically annotated documents, our approach operates in four steps, where only steps 3 and 4 are performed online, during similarity calculation of the query document:

1. **Expand query document:** Enrich query document with hierarchically and transversally related entities from a knowledge graph.
2. **Store expanded document:** Add expanded query document to existing corpus of documents so that, in future search queries, it can also be found as a result.
3. **Pre-search:** Use an inverted index to locate and rank documents that appear most related to the query based on entity overlap. Return a candidate set consisting of the top n ranking entries.
4. **Full search:** Determine pairwise similarity between query document and candidate documents on the annotation level. Rank candidates accordingly and return top k.

In terms of similarity notions, the pre-search step performs a rough preselection from the full body of entities in a document, independent of the specifics of how they tie into the document.

Reducing the number of documents to be processed to a significantly smaller number n, allows the application of the more granular, yet more expensive *full search*. Pairwise similarity scores between the query document and each candidate document are computed on the entity level to better capture sub-document and entity-level affiliations than by considering a document as a whole.

3.1 Entity Similarity

In this section, we describe the entity similarity measures that underlie document similarity (Sec. 3.2). Using the enriched annotations in semantically expanded documents, we are able to compute entity similarity metrics efficiently and without performing graph traversal. Analogous to document expansion, similarity computation is divided into hierarchical and transversal parts. To benefit of both, we combine transversal and hierarchical scores into one entity similarity score

$$sim_{ent}(e_1, e_2) = transSim^{norm}(e_1, e_2) + hierSim^{norm}(e_1, e_2)$$

using normalized (by mean and variance) versions of $hierSim$ and $transSim$, which we explain in the following.

Hierarchical entity similarity In hierarchical document expansion, each annotation gets enriched with the name and depth of its ancestor categories. We define **hierarchical entity similarity**

$$hierSim(e_1, e_2) = 1 - d(e_1, e_2)$$

using for d one of the two taxonomical distance measures **dps** [17] and **dtax** [4], as inspired by Palma et al.[15]. Both dps and dtax utilize the graph-theoretic concept of Lowest Common Ancestor (LCA). For nodes x and y, we define the LCA as a random representative from the subset of deepest nodes of x's and y's ancestors overlap.

Let $d(a, b) = |depth(a) - depth(b)|$, if a is an ancestor of b or vice versa, and $d(a, b) = 0$ otherwise. *dtax* follows the notion that the closer two nodes are to their LCA, compared to their distances to the root:

$$d_{tax}(x, y) = \frac{d(lca(x, y), x) + d(lca(x, y), y)}{d(root, x) + d(root, y)}$$

dps expresses distance in terms of the depth of two entities' LCA, compared to the entities' distances to their LCA:

$$d_{ps}(x, y) = 1 - \frac{d(root, lca(x, y))}{d(root, lca(x, y)) + d(lca(x, y), x) + d(lca(x, y), y)}$$

Transversal entity similarity Given two annotations a_1, a_2 and expansion radius L, we find paths of length up to $2 * L$ that connect them, then compute a score depending on the length and number of those paths. We first compute

$$trans(a_1, a_2) = \sum_{l=0}^{L*2} \beta^l * |paths^{(l)}_{(a_1,a_2)}|$$

with $paths^{(l)}_{(a_1,a_2)}$ the set of paths of length l based on outgoing edges from the annotations. The formula is inspired by Nunes et al.'s *Semantic Connectivity Score* in [14]. However, instead of finding paths through graph traversal, we use the weights assigned to all entities in a_1's and a_2's respective L-step transversal neighborhood during document expansion. Let $paths_{(a_1,e,a_2)}$ denote the concatenation of $paths_{(a_1,e)}$ and $paths_{(a_2,e)}$, i.e. all paths from either annotation that connect it to e. With a_i's neighborhood $N(a_i)$, it is $trans(a_1, a_2) =$

$$\sum_{l=0}^{L*2} \beta^l * |paths^{(l)}_{(a_1,a_2)}| = \sum_{e \in N(a_1) \cap N(a_2)} (\sum_{l=0}^{L*2} \beta^l * |paths^{(l)}_{(a_1,e,a_2)}|)$$

$$= \sum_{e \in N(a_1) \cap N(a_2)} ((\sum_{i=0}^{L} \beta^i * |paths^{(i)}_{(a_1,e)}|) * (\sum_{j=0}^{L} \beta^j * |paths^{(j)}_{(a_1,e)}|))$$

$$= \sum_{e \in N(a_1) \cap N(a_2)} w_{a_1}(e) * w_{a_2}(e)$$

This makes $trans_{a_1,a_2}$ easy to compute. Also, it is easily extendable to work with bidirectional expansion and paths: when considering edges of both directionalities, paths of length $> 2L$ will overlap on multiple nodes. This effect can be counteracted by penalizing path contribution depending on its length. Finally, we receive our **transversal similarity** measure by normalizing the score

$$transSim(a_1, a_2) = \frac{trans(a_1, a_2)}{trans(a_1, a_1)}$$

3.2 Document similarity

Analyzing pairwise relationships between two documents' annotations makes it possible to explicitly assess how each single annotation corresponds to another document. We regard two documents as similar if many of a documents' annotations are related to at least one annotation in the respective other document. In other words, given two documents, we want to connect each entity of both annotation sets with its most related counterpart. Unlike some other approaches[15][6], we do not aim for a 1-1 matchings between annotations - we argue that equal or similar concepts in two documents can be represented by varying numbers of annotations, in particular when using automatic entity extraction. Our approach works in three steps:

1. **Full bipartite graph**: for each annotation pair (a_1, a_2) (a_i: annotation of document i), compute entity similarity score.
2. **Reduce graph**: start with empty *maxGraph*. For each annotation, add adjacent edge with maximum weight to the *maxGraph*.
3. **Compute document score**:

$$sim_{doc}(d_1, d_2) = \frac{\sum_{a_{1i} \in A_1} (sim_{ent}(a_{1i}, matched(a_{1i})))}{|A_1| + |A_2|}$$

   with $matched(a_1), a_1 \in A_1$ denoting the annotation $a_2 \in A_2$ that a_1 has an edge to in *maxGraph*.

Figure 2 illustrates an example of our approach. While edges are displayed as undirected, each edge $e = (v, w)$ carries $sim_{ent}(v, w)$ close to e's end towards v, and vice versa at its end towards w.

3.3 Computational Complexity

Our indexing of expanded entities and documents as well as the resulting possibility of a pre-search step does no less than *enable* large-scale search applications to use graph-based similarity. When faced with millions of documents, the number of computations between entities of a query and all documents would soon become overwhelming.

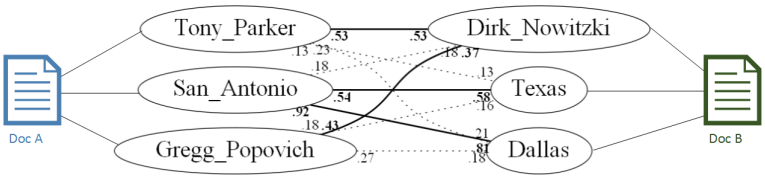


Fig. 2. Bipartite graph between sample documents 1 and 2. Bold lines constitute *max-Graph*

To compute pairwise entity similarity, any shortest-path-based algorithm ought to traverse all edges in the entities' neighborhoods in order to find connecting paths between them. For any subgraph G that is explored in the process, it holds that $|E| \leq \frac{|V|(|V|-1)}{2}$, i.e. that the maximum number of edges in E grows quadratically with the number of vertices in V. Another way of looking at this is that the number of edges that need to be traversed grows exponentially with the intended path length. In comparison, by traversing the graph and computing node scores at indexing time, we reduce this search-time complexity to be linear in $|V|$: the nodes of subgraph G can simply be retrieved and its node scores then be used in pairwise document similarity.

4 Related Work

Word Distribution based Document Similarity: Document search requires a method for efficient retrieval of relevant documents along with a scoring metric to rank candidates. Traditional text search approaches rely on the bag-of-words model and the distributional hypothesis. More sophisticated statistical approaches involve other sources of information in order to create more meaningful features in a document: Explicit Semantic Analysis(ESA) [9] represents text as a vector of relevant concepts. Each concept corresponds to a Wikipedia article mapped into a vector space using the TF-IDF measure on the article's text. Similarly, Salient Semantic Analysis (SSA) [10] use hyperlinks within Wikipedia articles to other articles as vector features, instead of using the full body of text.

While quick to compute, distributional metrics can perform poorly due to a lack of explicit information. Figure 3 demonstrates this for query "Gregg Popovich", coach of the San Antonio Spurs basketball team: while ESA ranks NBA players Bryant, Leonard, Nowitzki and Parker with no intuitive order, our knowledge-based method correctly recognizes a closer relationship between Gregg Popovich and his own (Spurs) players Kawhi Leonard and Tony Parker.

Graph based Document Similarity: [6] present one of several approaches for text similarity based on the lexical knowledge graph WordNet.[5] This measures similarity on a lexicographic level, whereas we are interested in conceptual semantic knowledge, as can be found in DBPedia.

[5] https://wordnet.princeton.edu/

Fig. 3. Entity similarity scores using ESA (left) and our knowledge-based similarity using DBPedia (right) for "Gregg Popovich".

Rank	Entity	Score
1	San Antonio	0.026
2	Kobe Bryant	0.013
3	Kawhi Leonard	0.006
4	Dirk Nowitzki	0.006
5	Tony Parker	0.006
6	Phil Jackson	0.004

Rank	Entity	Score
1	Tony Parker	0.921
2	Kawhi Leonard	0.827
3	San Antonio	0.644
4	Kobe Bryant	0.604
5	Phil Jackson	0.533
6	Dirk Nowitzki	0.506

Metrics such as PathSim[20] and HeteSim[19] assess the similarity of entities in heterogeneous graphs based on paths between them. Bhagwani et al. Leal et al.[11] and Lam et al.[13] suggest methods for measuring relatedness of DBPedia entities. In order to accommodate DBPedia's heterogeneity, Leal et al.'s approach accepts a domain configuration to restrict DBPedia to a subgraph; Lam et al. apply a TF-IDF-inspired edge weighting scheme and Markov Centrality to rank entities by similarities with respect to a query entity. Nunes et al.[14] present a DBPedia-based document similarity approach, in which they compute a document connectivity score based on document annotations. In a follow-up paper[5] they note by themselves that for the "Semantic Search" use case, they use traditional TF-IDF because their pairwise document similarity measure is too complex. All approach mentioned above differ from ours in several ways since they don't have a sophisticated hierarchical and transversal similarity metric.

Thiagarajan et al. [22] present a general framework how spreading activation can be used on semantic networks to determine similarity of groups of entities. They experiment with Wordnet and a Wikipedia Ontology as knowledge bases and determine similarity of generated user profiles based on a 1-1 annotation matching. In the Wikipedia Ontology, they restrict investigated concepts to parent categories. While our use of spreading activation is for transversal edges, we apply specialized taxonomical measures for hierarchical similarity. Palma et al. describe an annotation similarity measure *AnnSim* with which they evaluate similarity of interventions/drugs through biomedical annotations[15] using a 1-1 matching of annotations. We incorporated ideas from *AnnSim* into our hierarchical similarity measure.

Schuhmacher and Ponzetto's work [18] features entity and document similarity measures based on DBPedia entity linking and analysis of entity neighborhoods, making it particularly similar to our transversal similarity. However, they lack a notion for hierarchical similarity and their similarity metric differs in that it is based on *Graph Edit Distance*, and limits the maximum length of paths explored between entities to two, while we have successfully experimented with lengths of up to six (see $GBSS_3$ in sec. 5.2).

A major difference of all graph based approaches mentioned above relates to computational complexity. As discussed in Sec. 3.3, as graph traversal is performed at "query time".

5 Evaluation

Our evaluation aims at showing (i) how different settings influences the performance of our approach, (ii) that it can be be computed quickly, (iii) that our graph-based document similarity outperforms all related approaches for multiple-sentence documents (Sec. 5.2) and (iv) even single sentences as soon as they have at least one annotated entity (Sec. 5.3).

5.1 Experimental setup

We implemented our related-document search using the following resources:

- **DBPedia:** While the methods we present in this paper can be applied on any suitable semantic knowledge base, we choose DBPedia for our implementation because of its general-purpose, multilingual nature and comprehensiveness. Following Damljanovic et al.[8], we define the predicate types `skos:broader`, `rdf:type`, `rdfs:subclassOf` and `dcterms:subject` as hierarchical, while we consider semantic, non-hierarchical links from the *DBPedia Ontology* as transversal.
  Wikipedia Category Hierarchy: DBPedia contains multiple classification systems, namely YAGO, Wikipedia Categories and the hierarchical subgraph of the DBPedia Ontology. According to Lam et al.[13], the Wikipedia Category system has the highest coverage of entities among all three options. However, it does not have tree structure, plus various sources (e.g. [13], [16]) confirm that it contains cycles. To overcome these issues, we use the *Wikipedia Category Hierarchy* by Pavan et al.[16].
- **Lucene:** For the candidate document retrieval needed in the pre-search step, we leverage the built-in indexing capabilities of Lucene through *MoreLikeThis* queries. We store semantically expanded documents by adding two fields *transversal* and *hierarchical* to each document, for which we store term vectors: in each entry, the term represents the entity, while the term frequency captures the weight.
- **Jena:**[6] We use Jena TDB triplestores to operate DBPedia locally. We also store semantically expanded documents in a dedicated TDB store from where they can be efficiently retrieved at search time.
- **xLisa semantic annotator:**[23] We use xLisa to annotate text documents with DBPedia entities.

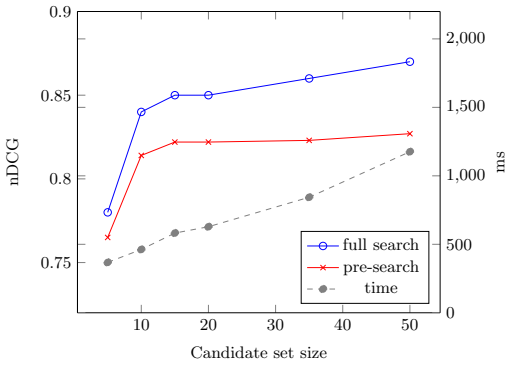
5.2 Comparing Mulitple-Sentence Documents

We use the standard benchmark for multiple sentence document similarity to evaluate how well our metrics approximate the human notion of similarity. This corpus was compiled by Lee et al. of 50 short news articles of between 51 to 126 words each with pairwise ratings of document similarities.[7].

[5] http://wiki.dbpedia.org/Datasets2014
[6] https://jena.apache.org/documentation/tdb
[7] Lee50 dataset available at https://webfiles.uci.edu/mdlee/LeePincombeWelsh.zip

Fig. 4. Left: Correlation (Pearson (r), Spearman (ρ) and their harmonic mean μ) and $nDCG$ ranking quality for different measures in our approach. Right: $nDCG$ for pre-search , full search, and execution time.

	Correlation			Ranking
	r	ρ	μ	$nDCG$
$TSS_{r=0}$	0.59	0.46	0.517	0.811
$TSS_{r=1}$	0.641	0.424	0.510	0.846
$TSS_{r=2}$	0.663	0.437	0.527	0.851
$TSS_{r=3}$	0.62	0.442	0.516	0.802
HSS_{dtax}	0.652	0.51	0.572	0.827
HSS_{dps}	0.692	0.511	0.588	0.843
$GBSS_{r=1}$	0.7	0.507	0.588	0.863
GBSS$_{r=2}$	**0.714**	0.511	0.596	**0.870**
GBSS$_{r=3}$	0.704	**0.519**	**0.598**	0.863



Semantic Similarity evaluation We assess Pearson and Spearman correlation plus their harmonic mean, as well as ranking quality using *Normalized Discounted Cumulative Gain (nDCG)* The stated correlation metrics are also used in related work and thus allow us to compare our results to other approaches. With nDCG, we aim at measuring how well *relevant* documents are discovered, which is an important criterion for the related-document search use case. To capture relevant documents only, we confine the quality evaluation to the top $m(q)$ elements. For query document q, it is defined as twice the number of documents that humans scored greater than or equal to 3.0.

nDCG scores reported in this section represent the average nDCG score obtained by using each Lee50 document as a query document once. Table 4 lists document similarity correlation and ranking quality based on the different measures we developed in our work:

- **Transversal Semantic Similarity (TSS):** Similarity score is solely based on transversal edges, as described in 3.1. The applied expansion radius is indicated in the subscript (e.g. $TSS_{r=2}$).
- **Hierarchical Semantic Similarity (HSS):** Similarity score is solely based on hierarchical edges, using one of the metrics *dps* or *dtax*.
- **Graph-based Semantic Similarity ($GBSS$):** Combination of TSS and HSS, as described at the top of Sec. 3.1. The subscript denotes the expansion radius used in transversal similarity assessment.

Table 4 shows that using *dps* in hierarchical similarity yields better results than *dtax*. Transversal similarity achieves peak performance for expansion radius set to two - interestingly, it fares very well for ranking quality, while falling behind on correlation. Upon closer examination, many transversal document similarities turned out to be zero: while hierarchically, even strongly unrelated entities tend to share features on some abstract level and thus yield a score greater than

zero, there is often no (short) transversal path between them. Moreover, results suggest that transversal paths longer than four or five (2∗ expansion radius) contain little value but add noise to the calculation.

By combining transversal and hierarchical (*dps*) scores for each entity in the $GBSS$ method, we achieved the best results across correlation and ranking quality. This demonstrates the different notions behind transversal and hierarchical similarity and that they can both add value to semantic measures.

Related-Document Search The plot in Figure 4 illustrates ranking quality after pre-search and full search as well as the average processing time per related-document search. We chose $GBSS_{r=2}$ as similarity metric in the full search step because it performed best on ranking quality. The plot shows that generally, the larger the candidate set, the better the quality. The fact that full search achieves higher nDCG scores than pre-search confirms the successful re-ordering that takes place in full search based on pairwise entity-based similarity computation.

Except for pre-search, which is performed offline, our approach's speed is independent of corpus size and only depends on candidate set size: the gray line shows that processing time grows linearly with candidate set size and confirms the efficiency of our search approach. Table 5 breaks down the total execution time into its elements: given semantically expanded documents, the pairwise similarity computations in full search prove to be very fast. The bottleneck of our implementation turns out to be the retrieval of semantically expanded documents from a Jena TDB; using different means of storage, this should be easy to improve.

Comparison to related work Table 5 lists the performance for our two best-performing similarity measures $GBSS_{r=2}$ and $GBSS_{r=3}$, as well as for the following related approaches:

- **TF-IDF:** Distributional baseline algorithm.
- **AnnOv:** Similarity score based on annotation overlap. Corresponds to $TSS_{r=0}$ in table 4.
- **Explicit Semantic Analysis (ESA):** Via the public ESA REST endpoint,[8] we computed pairwise similarities for all document pairs.
- **Graph Edit Distance (GED):** correlation value for GED was taken from Schuhmacher [18]
- **Salient Semantic Analysis (SSA), Latent Semantic Analysis (LSA):** correlation values for SSA and LSA were taken from Hassan and Mihalcea[10].

Table 5 clearly shows that our approach significantly outperforms the to our knowledge most competitive related approaches, including Wikipedia-based SSA and ESA. While ESA achieves a rather low Pearson correlation and SSA comparably low Spearman correlation, our approach beats them in both categories. To

[8] http://vmdeb20.deri.ie:8890/esaservice

Fig. 5. Left: Correlation (Pearson (r), Spearman (ρ) and their harmonic mean μ) in comparison with related work. Right: Processing times for elements of our related-document search (candidate set size: 50)

		Correlation		
		r	ρ	μ
Baseline	$TF-IDF$	0.398	0.224	0.286
	$AnnOv$	0.59	0.46	0.517
Related	LSA	0.696	0.463	0.556
	SSA	0.684	0.488	0.569
	GED	0.63	-	-
	ESA	0.656	0.510	0.574
Ours	$\mathbf{GBSS}_{r=2}$	**0.712**	0.513	0.596
	$\mathbf{GBSS}_{r=3}$	0.704	**0.519**	**0.598**

Operation	Time(ms)
1 document expansion	121.48
Generate candidate set (size 50)	51.0
Retrieve 50 expanded documents	794.42
50 sim_{doc} computations	209.18
Total time	1176.08

compare ranking quality, we also computed nDCG for the best-scoring related approach ESA, where it reaches 0.845: as table 4 shows, our approach scores also beats that number significantly.

5.3 Comparing Sentences

Since we base similarity on annotations our approach requires documents for which high-quality annotations can be produced. Entity linking tools like xLisa[23] tend to produce better results with longer document (several sentences or more). To investigate if our approach is also competitive on very short documents like single sentences we performed experiments on the SemEval task about Semantic Textual Similarity (henceforth STS)[9]. Each document only contains around 8-10 words and around 1-3 entities.

Datasets: We picked two datasets in the STS task from different years that have a good representation of entities. We considered those sentences with at least one linkable entity mention.

- **2012-MSRvid-Test:** It is collected from the MSR Video Paraphrase Corpus[10] that provides sentence description about the event in a video. For the SemEval-2012[11] task, 750 pairs of sentences for training and 750 for testing were used. We used the dedicated testing part of the dataset on which baseline scores were reported.
- **2015-Images:** It is collected as a subset of the larger Flickr dataset containing image descriptions. The dataset consists of around 750 sentence pairs.

[9] http://ixa2.si.ehu.es/stswiki/index.php/Main_Page
[10] http://research.microsoft.com/en-us/downloads/38cf15fd-b8df-477e-a4e4-a4680caa75af/
[11] https://www.cs.york.ac.uk/semeval-2012/task6/index.html

Competing Approaches: To have a fair comparison, we only report those related approaches that perform unsupervised semantic similarity assessment, meaning they don't exploit the given manual similarity assessment to train or optimize the model, but only use these target values for evaluation of the approach.

- **STS-12:** Baseline approach reported in SemEval-2012 [2] for MSRvid-Test dataset.
- **STS-15:** Baseline approach reported in SemEval-2015 [1] for Images dataset.
- **Polyglot:** Word embeddings obtained from Al-Rfou et al. [3] is used to calculate the sentence embeddings by averaging over word embeddings. Sentence words that are not observed in the word embedding database are ignored.
- **Tiantianzhu7:** Uses a graph-based word similarity based on word-sense disambiguation.
- **IRIT:** Uses a n-gram comparison method combined with WordNet to calculate the semantic similarity between a pair of concepts.
- **WSL:** Uses a edit distance to include word order by considering word context.

Results: Table 1 shows the results based on Pearson correlation (r) values. Again, we clearly outperform related approaches with both types of document representations, graph-based and word-distribution-based. This indicates, that as long as we obtain one correct entity to represent a document our sophisticated hierarchical and transversal semantic similarity measure can compete with the state-of-the-art even for very short text.

Table 1. Correlation (Pearson (r) in comparison with related work.

		Sentence Semantic Similarity	
		2012-MSRvid-Test	2015-Images
Baseline	STS-12	0.299	-
	STS-15	-	0.603
Related	Polyglot [3]	0.052	0.194
	Tiantianzhu7 [24]	0.594	-
	IRIT [7]	0.672	-
	WSL [21]	-	0.640
Ours	**GBSS**$_{r=2}$	0.666	**0.707**
	GBSS$_{r=3}$	**0.673**	0.665

6 Conclusion

In this paper, we have presented a new approach for efficient knowledge-graph-based semantic similarity. Our experiments on the well-established Lee50 document corpus demonstrate that our approach outperforms competing approaches

in terms of ranking quality and correlation measures. We even achieve superior performance for very short documents (6-8 words in the SemEval task) as long as we can link to at least one entity. By performing all knowledge graph-related work in the *Semantic Document Expansion* preprocessing step, we also achieve a highly scalable solution.

The strong performance of our similarity measure demonstrates that semantic graphs, including automatically generated ones like DBPedia contain valuable information about the relationship of entities. Moreover, similarity measures can be developed that compete with traditional word-distribution based approaches in every aspect.

For future work, testing on diverse corpora with documents differing in language, vocabulary and modality seems promising.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 611346.

References

1. Agirre, E., Banea., C.: Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. Association for Computational Linguistics (2015)
2. Agirre, Eneko, M.D.D.C., Gonzalez-Agirre., A.: Semeval-2012 task 6: A pilot on semantic textual similarity. pp. 385–393. Association for Computational Linguistics, Sofia, Bulgaria (2012)
3. Al-Rfou, R., Perozzi, B., Skiena, S.: Polyglot: Distributed word representations for multilingual nlp. pp. 183–192. Association for Computational Linguistics, Sofia, Bulgaria (August 2013)
4. Benik, J., Chang, C., Raschid, L., Vidal, M.E., Palma, G., Thor, A.: Finding cross genome patterns in annotation graphs. In: Data Integration in the Life Sciences, Lecture Notes in Computer Science, vol. 7348, pp. 21–36. Springer Berlin Heidelberg (2012)
5. Bernardo Pereira Nunes and Besnik Fetahu and Marco Antonio Casanova: Cite4me: A semantic search and retrieval web application for scientific publications. LAK (Data Challenge)'13 (2013)
6. Bhagwani, S., Satapathy, S., Karnick, H.: Semantic textual similarity using maximal weighted bipartite graph matching. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation. pp. 579–585. SemEval '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012)
7. Buscaldi, D., T.R.A.G.N., Mothe, J.: Irit: Textual similarity combining conceptual similarity with an n-gram comparison method. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation. pp. 552–556. Association for Computational Linguistics (2012)

8. Damljanovic, D., Stankovic, M., Laublet, P.: Linked data-based concept recommendation: Comparison of different methods in open innovation scenario. In: Proceedings of the 9th International Conference on The Semantic Web: Research and Applications. pp. 24–38. ESWC'12, Springer-Verlag, Berlin, Heidelberg (2012)
9. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: IJCAI. vol. 7, pp. 1606–1611 (2007)
10. Hassan, S., Mihalcea, R.: Semantic relatedness using salient semantic analysis. In: AAAI (2011)
11. Jose Paulo Leal, Vania Rodrigues, Ricardo Queiros: Computing semantic relatedness using dbpedia. In: 1st Symposium on Languages, Applications and Technologies. OpenAccess Series in Informatics (OASIcs), vol. 21, pp. 133–147. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2012)
12. Katz, L.: A new status index derived from sociometric analysis. Psychometrika 18(1), 39–43 (1953)
13. Lam, S., Hayes, C., DERI, N.U., Park, I.B.: Using the structure of dbpedia for exploratory search. In: KDD '13: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA (2013)
14. Nunes, B.P., Kawase, R., Fetahu, B., Dietze, S., Casanova, M.A., Maynard, D.: Interlinking documents based on semantic graphs. Procedia Computer Science 22, 231–240 (2013)
15. Palma, G., Vidal, M.E., Haag, E., Raschid, L., Thor, A.: Measuring relatedness between scientific entities in annotation datasets. In: Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics. pp. 367:367–367:376. BCB'13, ACM, New York, NY, USA (2013)
16. Pavan Kapanipathi, Prateek Jain, Chitra Venkataramani, Amit Sheth: Hierarchical interest graph (21 January 2015), http://wiki.knoesis.org/index.php/Hierarchical_Interest_Graph
17. Pekar, V., Staab, S.: Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. In: Proceedings of the 19th international conference on Computational linguistics-Volume 1. pp. 1–7 (2002)
18. Schuhmacher, M., Ponzetto, S.P.: Knowledge-based graph document modeling. In: WSDM. pp. 543–542. ACM (2014)
19. Shi, C., Kong, X., Huang, Y., Philip, S.Y., Wu, B.: Hetesim: A general framework for relevance measure in heterogeneous networks. IEEE Transactions on Knowledge & Data Engineering (10), 2479–2492 (2014)
20. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. VLDB'11 (2011)
21. Takagi, N., Tomohiro., M.: Wsl: Sentence similarity using semantic distance between words. In: SemEval. Association for Computational Linguistics (2015)
22. Thiagarajan, R., Manjunath, G., Stumptner, M.: Computing semantic similarity using ontologies. In: ISWC 08, the International Semantic Web Conference (ISWC) (2008)
23. Zhang, L., Rettinger, A.: X-lisa: Cross-lingual semantic annotation. Proceedings of the VLDB Endowment (PVLDB), the 40th International Conference on Very Large Data Bases (VLDB) 7(13), 1693–1696 (September 2014)
24. Zhu, T., Lan., M.: System description of semantic textual similarity (sts) in the semeval-2012 (task 6). In: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation. Association for Computational Linguistics (2012)