Linked Data Enabled Generalized Vector Space Model To Improve Document Retrieval

Jörg Waitelonis, Claudia Exeler, and Harald Sack

Hasso-Plattner-Institute for IT-Systems Engineering, Prof.-Dr.-Helmert Str. 2-3,
14482 Potsdam, Germany
joerg.waitelonis@hpi.de, claudia.exeler@student.hpi.de,
harald.sack@hpi.de

Abstract. This paper presents two approaches to semantic search by incorporating Linked Data annotations of documents into a Generalized Vector Space Model. One model exploits taxonomic relationships among entities in documents and queries, while the other model computes term weights based on semantic relationships within a document. We publish an evaluation dataset with annotated documents and queries as well as user-rated relevance assessments. The evaluation on this dataset shows significant improvements of both models over traditional keyword based search.

1 Introduction

Due to the increasing demands of information seekers, retrieving exactly the right information from document collections is still a challenge. Search engines try to overcome the drawbacks of traditional keyword based search systems, such as ambiguities of natural language (vocabulary mismatch), by the use of knowledge bases, as e. g. Google's Knowledge Graph[1]. These knowledge bases enable the augmentation of search results with structured semantic information gathered from various sources. With these enhancements, users can more easily explore the result space and satisfy their information needs without having to navigate to other web sites [22, 16]. Linked Open Data (LOD) provides numerous structured knowledge bases, but there are even more ways in which LOD can improve search.

Many information needs go beyond the retrieval of facts. Full documents with comprehensive textual explanations have much greater power to provide an actual understanding than any structured information will ever have. On the web, there are relevant documents for almost any imaginable topic. Thus, to find the right documents is a matter of accurately specifying the query keywords. Typically, users start with a general query and refine it when the search results do not contain the expected result. For most cases this process works fine, because any query string matches at least some relevant documents. The challenge of web

[1] http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html

search is thus to determine the highest quality results among a set of matching documents. In contrast to web search, query refinement can quickly lead to empty result sets in document collections of limited size, such as blogs, multimedia archives, or libraries, because the wrong choice of keywords may eliminate the only relevant document. One approach to cope with these shortcomings is to explicitly map the document content to entities of a formal knowledge base, and to exploit this information by taking into account semantic similarity as well as relatedness of documents and queries.

For this purpose, we have developed and comprehensively evaluated a *Semantic Search* system, which combines traditional keyword based search with LOD knowledge bases, in particular DBpedia[2]. Our approach shows that retrieval performance on less than web-scale search systems can be improved by the exploitation of graph information from LOD resources. The main contributions of this paper are two novel approaches to exploit LOD knowledge bases in order to improve document search and retrieval based on:

1. the adaption of the generalized vector space model (GVSM) with *taxonomic relationships*,
2. measuring the level of *connectedness* of entities within documents instead of traditional term frequency weighting,

Furthermore, we have published a manually assembled and carefully verified evaluation data set with semantically annotated documents, search queries, as well as relevance assessments at different relatedness levels[3].

In the remainder of this paper, Section 2 provides a technical instruction and references related work. Section 3 describes the two proposed approaches and Section 4 addresses their evaluation. The final section summarizes and discusses the achieved results, and gives a brief outlook on future work.

2 Preliminaries and Related Work

Semantic Search makes use of explicit semantics to solve core search tasks, i. e. interpreting queries and data, matching query intent with data, and ranking search results according to their relevance for the query. In modern semantic retrieval systems the ranking also makes use of underlying knowledge bases to obtain the degree of *semantic similarity* between documents and queries [6]. One of the most popular retrieval models to determine similarity between documents and queries is the *Vector Space Model (VSM)*. Basically, it assumes pairwise orthogonality among the vectors representing the index terms, which means that index terms are independent of each other. Hence, VSM does not take into account that two index terms can be semantically related. Therefore, Wong et. al. have introduced the *Generalized Vector Space Model (GVSM)*, where the index terms are composed of smaller elements and term vectors are not considered

[2] http://dbpedia.org/

[3] The ground truth dataset is published at: http://s16a.org/node/14

pairwise orthogonal in general [23]. The similarity function which determines the similarity among documents d_k and the query q is extended with a term correlation $\boldsymbol{t_i} \cdot \boldsymbol{t_j}$:

$$sim_{cos}(\boldsymbol{d_k}, \boldsymbol{q}) = \frac{\sum_{j=1}^{n} \sum_{i=1}^{n} d_{k,j} q_j \boldsymbol{t_i} \boldsymbol{t_j}}{\sqrt{\sum_{i=1}^{n} d_{k,i}^2} \sqrt{\sum_{i=1}^{n} q_i^2}}, \tag{1}$$

where $d_{k,i}$, q_i represent the weights in the document and query vectors, n the dimension of the new vectors. The term correlation $\boldsymbol{t_i} \cdot \boldsymbol{t_j}$ can be implemented in different ways. Wong et. al. have used co-occurrences [23] and the model in [17] uses WordNet, a large lexical database of words grouped into sets of synonyms (synsets), each expressing a distinct concept [11]. Our approach instead utilizes LOD resources and their underlying ontologies to determine a correlation between related index terms.

Before exploiting the semantic relatedness, the document contents must be annotated via *Named Entity Linking* (NEL). NEL is the task of identifying mentions of named entities in a text and linking them to the corresponding entities in a knowledge base. A wide range of approaches for NEL exists and most of them integrate natural language processing, such as named entity recognition, co-reference resolution, and word sense disambiguation (WSD) with statistical, graph-based, and machine learning techniques [19].

Our retrieval approach is inspired by the idea of *Concept-based document retrieval*, which uses WSD to substitute ambiguous words with their intended unambiguous concepts and then applies traditional IR methods [5]. Several knowledge bases have been exploited to define concepts. One of the first concept-based IR approaches in [21] uses the WordNet taxonomy, whereas the more recent Explicit Semantic Analysis [3, 4] is based on concepts that have been automatically extracted from Wikipedia.

Some approaches have already attempted to include semantic relationships in a retrieval model. Lexical relationships on natural language words have been applied by [7] for query expansion and by [17] in a GVSM. However, their lack of disambiguation introduces a high risk for misinterpretation and errors. The latter model nevertheless shows small improvements, but is limited by the knowledge represented in WordNet. The fact that named entities play an important role in many search queries has been considered by [1, 12]. Their approach focuses on correctly interpreting and annotating of the query and extending the query with names of instances of found classes.

Another approach is applied by [20, 2]. They use formal SPARQL[4] queries to identify entities relevant to the user's information need and then retrieve documents annotated with these entities. Since this requires knowledge of a formal query language, it is not suited for the ordinary user.

None of the above approaches provides an allround service for end-user centered semantic search, which simultaneously builds on a theoretically sound retrieval model and is proven to be practically useful. Neither do any of them take

[4] `http://www.w3.org/TR/sparql11-overview/`

advantage of the relationships of concepts represented in a document. These are also the main points that the approaches presented in the following sections address.

3 Linked Data Enabled GVSM

With the goal to increase search recall, the *taxonomic approach* uses taxonomic relationship within the knowledge base to determine documents containing entities that are not explicitly mentioned in, but strongly related to the query. We go beyond any of the previous GVSM approaches by exploiting the semantic relationships to also identify documents that are not directly relevant, but related to the search query. Related documents serve as helpful recommendations if none or only few directly relevant documents exist, which is a frequent scenario when searching in limited document collections. Furthermore, taxonomies provide subclass relationships necessary for effectively answering class queries, a special kind of topical searches, where any members of a class are considered relevant. For example, the class search query “Tennis Players” should also return documents about instances of the class “Tennis Players”, such as “Andrew Agassi”, “Steff Graf”, etc., even if the term or entity “Tennis Players” does not explicitly occur in these documents.

In addition to the taxonomic approach, we also propose a *connectedness approach*, which aims at increasing search precision. In general, this approach computes an improved term weighting by analyzing semantic relationships between the entities within a document.

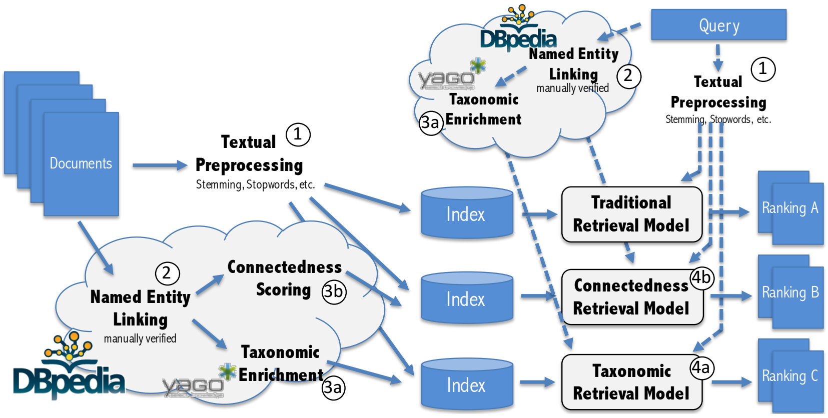


Fig. 1. Evaluation architecture overview.

Fig. 1 shows the entire workflow of both proposed semantic search approaches in addition to traditional keyword based search. The workflow consists of four processing steps: (1) traditional syntactic document and query processing, (2) semantic document and query annotation with LOD resources, (3) annotation enrichment with semantic information, and (4) query to document matching and result ranking. These steps have been implemented into the Apache SOLR/Lucene[5] indexing and retrieval system. Their purpose, theoretical background and implementation are explained in detail in the following.

The process starts with textual preprocessing (step 1), which applies stopword removal, basic synonym expansion, and word stemming to the document texts. The resulting textual index terms constitute one part of the index for both proposed approaches, so that textual and semantic index terms are treated equally. In parallel, step 2 performs semantic document annotation by NEL with DBpedia entities. Because NEL does not always guarantee correct results, the annotation of documents has been manually revised with [16].

3.1 Taxonomic Enrichment & Retrieval

The taxonomic approach is a variation of the GVSM. In a GVSM, term vectors are not necessarily orthogonal to reflect the notion that some entities or words are more closely related or similar to each other. This section describes the construction of term vectors from a taxonomy (step 3a) and the derived retrieval model for matching documents to a query (step 4a).

For index term associated with an entity, the term vectors $\boldsymbol{t_i}$ are constructed from the entity vector $\boldsymbol{e_i}$ of the entity it represents and the set of its classes $c(e_i)$:

$$\boldsymbol{t_i} = \alpha_e \boldsymbol{e_i} + \alpha_c \frac{\boldsymbol{v_i}}{|\boldsymbol{v_i}|}, \text{ with } \boldsymbol{v_i} = \sum_{c_j \in c(e_i)} w(c_j, e_i) \times \boldsymbol{c_j}. \tag{2}$$

The $\boldsymbol{e_i}$ and $\boldsymbol{c_j}$ are pairwise orthogonal vectors with n dimensions, where each dimension stands for either an entity or a class. Accordingly, n is the sum of the number of entities and the number of classes in the corpus. Taking the $\boldsymbol{c_j}$ to be orthogonal suggests that the classes are mutually independent, which is not given since membership in one class often implies membership in another. This model is thus not suited to calculate similarity between classes, but it does provide a vector space in which entities and documents can be represented in compliance with their semantic similarities.

The factors α_e and α_c determine how strongly exact matches of the query entities are favored over matches of similar entities. They are constant across the entire collection to make sure that the similarity of the term vectors corresponding to two entities with the same classes is uniform. To keep $\boldsymbol{t_i}$ a unit vector, they are calculated on a single value $\alpha \in [0, 1]$:

$$\alpha_e = \frac{\alpha}{\sqrt{\alpha^2 + (1-\alpha)^2}} \text{ and } \alpha_c = \frac{1-\alpha}{\sqrt{\alpha^2 + (1-\alpha)^2}}. \tag{3}$$

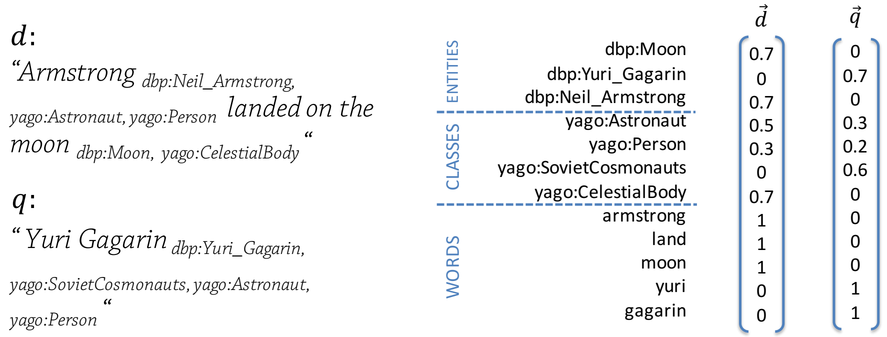
With higher α, documents with few occurrences of the queried entity will be preferred over documents with many occurrences of related entities.

[5] http://lucene.apache.org/

Since not every shared class means the same level of similarity between two entities, not all classes should contribute equally strong to the similarity score. Assigning weights $w(c_j, e_i)$ to the classes within a term vector achieves this effect. Without them, the cosine similarity of two document (or query) vectors solely depends on the number of classes shared by the entities they contain. The $w(c_j, e_i)$ should express the relevance of the class c_j to the entity e_i. We found that Resnik's relatedness measure [13], i.e. $w_{Resnik} = max_{c' \in S(c_j, e_i)} IC(c')$ performed best in our evaluations (cf. Sec. 4). $IC(c')$ expresses the specificity, or information content, of the class c', and can be calculated by measures like linear depth, non-linear depth [14], or Zhou's IC [24]. It is a valuable component for our approach because generic classes hold less information. For example, `British Explorers` is a more precise description of `James Cook` than the general class `Person`. Other similarities that include IC have been proposed in [9], [10], and [18].

For our implementation, we have employed the classes from YAGO[6], a large semantic knowledge base derived from Wikipedia categories and WordNet, which interlinks with DBpedia entities [15]. Its taxonomy, which we have extend with `rdf:type` statements to include the instances, is well suited for the taxonomic approach for two reasons. First, it is fine-grained and thereby also allows for a fine-grained determination of similar entities. Second, since the main taxonomic structure is based on WordNet, it has high quality and consistency. This is advantageous when using the taxonomy tree for similarity calculations. Other taxonomies, such as the DBpedia Ontology[7] or Umbel[8], also qualify.

The text index integrates into the model by appending the traditional document vector to the entity-based document vectors. Fig. 2 shows an example document d and query q annotated with entities (*dbp:*) and classes (*yago:*), as well as the corresponding document vector $\boldsymbol{d}$ and query vector $\boldsymbol{q}$.


d:
"Armstrong dbp:Neil_Armstrong, yago:Astronaut, yago:Person landed on the moon dbp:Moon, yago:CelestialBody"
q:
"Yuri Gagarin dbp:Yuri_Gagarin, yago:SovietCosmonauts, yago:Astronaut, yago:Person"
ENTITIES
CLASSES
WORDS

	$\vec{d}$	$\vec{q}$
dbp:Moon	0.7	0
dbp:Yuri_Gagarin	0	0.7
dbp:Neil_Armstrong	0.7	0
yago:Astronaut	0.5	0.3
yago:Person	0.3	0.2
yago:SovietCosmonauts	0	0.6
yago:CelestialBody	0.7	0
armstrong	1	0
land	1	0
moon	1	0
yuri	0	1
gagarin	0	1

Fig. 2. Example document and query vectors in the taxonomic model.

[6] `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/`

[7] `http://wiki.dbpedia.org/services-resources/ontology`

[8] `http://www.umbel.org`

3.2 Connectedness Approach

To determine the relevance of a term within a document, term frequency (tf) is not always the most appropriate indicator. Good writing style avoids word repetitions, and consequently, pronouns often replace occurrences of the actual term. However, the remaining number of occurrences of the referred-to word depends on the writer. In documents with annotated entities, term frequency is especially harmful when annotations are incomplete. Such incompleteness might result, for example, when only the first few occurrences of an entity are marked, annotations are provided as a duplicate free set on document level, or not all surface forms of an entity are recognized.

We therefore suggest to replace the term frequency weight by a new measure, *connectedness*, which requires adaptations of indexing (step 3b in Fig. 1) and similarity scoring (step 4b). The connectedness of an entity within a document is a variation of the degree centrality based on the graph representation of the underlying knowledge base [8]. It describes how strongly the entity is connected within the subgraph of the knowledge base induced by the document (*document subgraph).* This approach has the desirable side effect that wrong annotations tend to receive lower weights due to the lack of connections to other entities in the text.

The document subgraph D, as illustrated by the example in Fig. 3 (A), includes all entities that are linked within the document (e_1 to e_8) as well as all entities from the knowledge base that connect at least two entities from the document (e_9 to e_{10}). As connectedness is defined on undirected graphs, the function $rel(e_i, e_j)$ is applied to to create an undirected document subgraph. It returns true if and only if there exists some relation from e_i to e_j or from e_j to e_i. Each entity $e_i \in D$ has a set E_i of directly connected entities and a set F_i of indirectly connected entities:

$$E_i = \{e \in D | rel(e, e_i)\} \text{ and } F_i = \{e \in D | \exists x : rel(e, x) \wedge rel(x, e_i)\} \qquad (4)$$

Fig. 3 (B) illustrates E_i and F_i for the example document in part A.

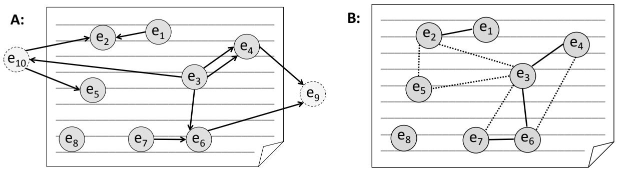


Fig. 3. A: Subgraph of a knowledge base spanned by a document. An arrow from entity e_i to e_j indidcates that an RDF triple $\langle e_i \rangle$ rel $\langle e_j \rangle$ exists in the underlying knwledge base. B: Each entity is connected to the members of its E_i by a solid line, and to the members of its F_i by a dashed line.

Based on these sets, connectedness $cn(e_i, d)$ is computed as follows:

$$cn(e_i, d) = 1 + (|E_i| + |F_i|) \times \frac{|D|}{n_d}, \text{ where } n_d = \sum_{e_j \in D} |E_j| + |F_j|. \tag{5}$$

Entities may have no connections to any other entities in the document subgraph. Since they are nevertheless relevant to the document, we add 1 to all scores. The multiplication with $|D|/n_d$ normalizes the score by the average number of connected entities over all $e \in D$. This normalization creates comparability between different documents, which is otherwise lacking for two reasons: On the one hand, entities are more likely to be connected to other entities in documents with more annotations. On the other hand, a single connection to another entity is more significant in a sparse document subgraph than in a dense one. The average score is preferable over the sum of scores because it keeps a larger variation in scores.

While calculating the term's weight within a document via connectedness, we keep the traditional inverse document frequency (*idf*) to calculate the term's distinctness. Whether or not a word or entity has a large power to distinguish relevant from non-relevant documents depends on the document corpus. In a corpus with articles about Nobel Prize winners, for example, 'Nobel Prize' is a common term, whereas in general collections, the same term is much less frequent, and its occurrence is more informative. Distinctness can thus only be accurately estimated by a corpus dependent measure.

Since connectedness is independent of taxonomic classes, for this approach the term vectors consist of the entity vector (all 0 for unannotated terms) concatenated with the traditional term vector. While weights in the traditional term vector part remain tf-idf weights, the entity vectors' values are cn-idf values, i.e.

$$w(e_i, d) = cn(e_i, d) \times idf(e_i) = \left(1 + (|E_i| + |F_i|) \times \frac{|D|}{n_d}\right) \times log\frac{|N|}{df(t)}. \tag{6}$$

On the other hand, connectedness is not suitable for weighting query entities. The main reason for this is that most times queries are too short to contain sufficiently many connections to convey any meaningful context. Furthermore, whether query weights are recommendable is application-dependent, and consequently left out in our considerations.

4 Evaluation

The evaluation shows how the proposed retrieval models improve the search effectiveness. To perform an initial optimization, we have manually assembled a ground truth dataset with documents, queries, and relevance judgements. Since human relevance judgements are idiosyncratic, variable and subjective, we have subsequently conducted a multi-user study with different judges, to double check whether the proposed method also performs well in a real user scenario.

An appropriate evaluation of the presented methods necessitates a correctly annotated dataset. This prohibits the use of traditional retrieval datasets, such as large scale web-search datasets, e. g. as provided by the TREC[9] community, because the semi-automatic creation of necessary semantic annotations would have taken too long. Datasets for NEL evaluation are perfectly annotated, but do not provide user queries and relevance judgments [19]. Hence, we decided to compile a new dataset. It consists of 331 articles from the yovisto blog[10] on history in science, tech, and art. The articles have an average length of 570 words, containing 3 to 255 annotations (average 83) and have been manually annotated with DBpedia entities [16]. Inspired by the blog's search log, we have assembled and also manually annotated a set of 35 queries. The blog authors, as domain experts, assisted in the creation of relevance judgements for each query.

On this initial dataset, we optimized some parameters of our approaches with respect to Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). The Resnik similarity with Zhou's IC performed best for the class-based method, and connectedness worked best when average normalization (as described in Section 3.2) was applied. For text-only search on this dataset, length normalization was not beneficial, and the classical linear term frequency weighting performed better than Lucene's default, which takes the square root of the term frequency.

With this configuration, we have set up a user study. Since ranking all 331 documents for 35 queries was beyond our capacities, we have used pooling to identify potentially relevant documents. For every query, the top 10 ranked documents from text-, class- (Resnik-Zhou), and connectedness search have been collected and presented to the users in random order. The users were then asked to assign every document to one of the following five categories based on its relation to the query: Document is relevant (5), parts are relevant (3), document is related (3), parts are related (1), irrelevant (0). The rounded arithmetic mean of the relevance scores (indicated in parentheses) determines all relevance score in the ground truth. Furthermore, the participants directly compared the rankings produced by text search (baseline), class-based search, and connectedness-based search and identified the best (score 2) as well as the second-best ranking (score 1). In total, 64 users have participated in the relevance assessments. All queries have been assessed by at least 8 participants.

Tab. 1 shows that the inclusion of semantic annotations and similarities clearly improves retrieval performance compared to traditional text search. As expected, the taxonomic approach increases overall recall (97.5% as compared to 93.2% for text search) because it is able to retrieve a larger number of documents. However, it also improves the ranking, measured by MAP and NDCG.

The connectedness approach performs better than text search, but worse than the other semantic methods, including the simple "Concept+Text" approach, where entities are treated as regular index terms within Lucene's default model. This poor performance is surprising because the results from the users'

[9] `http://trec.nist.gov/`

[10] `http://blog.yovisto.com/`

Table 1. Evaluation Results

Method	MAP	NDCG	MAP@10	NDCG@10	RR	Prec@1
Text (baseline)	0.696	0.848	0.555	0.743	0.960	0.943
Concept+Text ($\alpha = 1$)	0.736	0.872	0.573	0.761	0.979	**0.971**
Connectedness (only)	0.711	0.862	0.567	0.752	**0.981**	**0.971**
Connectedness (with tf)	0.749	0.874	0.583	**0.766**	0.979	0.943
Taxonomic (no similarity, $\alpha = \frac{1}{2}$)	0.766	0.875	0.603	0.758	0.961	0.943
Taxonomic (Resnik-Zhou, $\alpha = \frac{1}{2}$)	**0.768**	**0.877**	**0.605**	0.762	0.961	0.943

direct comparison of the three rankings indicates that connectedness (with an average score of 1.09) provides better rankings than taxonomic (1.01) and text search (0.90). This seeming contradiction hints at a difference between information retrieval evaluation measures and user perception of ranking quality. The evaluators have judged mainly by the very top few documents. Connectedness outperforms the other approaches in this respect, as shown by the reciprocal rank (RR) and precision@1 in Tab. 1. Also, connectedness performs best when related documents are not considered relevant.

Combining the connectedness measure with tf weights seems to be the best weighting. When considering only the first 10 documents, as users often do, this combined weighting's performance can be considered slightly better than the taxonomic approach due to its higher NDCG value, because NDCG takes different relevance levels into account, while MAP does not. However, connectedness is inferior to the taxonomic approach on the complete search results, because it simply does not retrieve certain documents. This harms the recall and negatively effects MAP and NDCG, while precision may still be higher.

Table 2. Comparison of semantic similarities for the taxonomic model

Similarity	Uniform	Jiang-Conrath [9]			Lin [10]		
Specificity	–	lin depth	log depth	Zhou's IC	lin depth	log depth	Zhou's IC
MAP	0.766	0.753	0.766	0.767	0.767	0.766	0.767
NDCG	0.875	0.868	0.875	0.875	0.876	0.875	**0.877**

Similarity	Resnik [13]			Tversky Ratio [18]	Tversky Contrast [18]
Specificity	lin depth	log depth	Zhou's IC	–	–
MAP	**0.768**	**0.768**	**0.768**	**0.768**	0.763
NDCG	**0.877**	0.876	**0.877**	0.876	0.873

Tab. 1 indicates that the use of the Resnik-Zhou similarity only has a very small positive impact on retrieval compared to the same approach with uniform weights. This is in line with the numbers from Tab. 2, which demonstrate that the choice of semantic similarity has only little impact. Apparently, the number of shared classes has a more significant influence in this setup than the class weights.

To determine the influence of NEL quality, we have also executed experiments with documents, where annotations have not been revised after automated NEL. The results still show an improvement over text search, but a MAP is 2.5% to 4.5% and NDCG 0.1% to 1.6% lower than in the equivalent experiments on manually revised documents.

5 Conclusions and Future Work

This paper has shown how Linked Data can be exploited to improve document retrieval based on an adaption of the GVSM. We have introduced two novel approaches utilizing taxonomic as well as connectedness features of Linked Data resources annotated within documents. Our evaluation has shown that both methods achieve a significant improvement compared to traditional text retrieval. The connectedness approach tends to increase precision whereas the taxonomic approach raises recall. Thereby, the similarity measure that weights taxonomic classes of index terms has only little influence on the retrieval. The quality of annotations substantially impacts the retrieval results, but uncorrected state-of-the-art NEL still ascertains an improvement. In general, the quality of Linked Data itself is an obstacle. Only about 65% of entities in the dataset used have type statements, which are essential for the performance of the taxonomy-aided retrieval. Since no annotated ground truth datasets with relevance judgements exist, we have created one via a pooling method.

There are still open questions, which are to be answered in future work, including how well the models perform with other knowledge bases (e. g. Wikidata or national authority files), what other semantic relations between entities are valuable for document retrieval, and how the semantic similarity can obtain more influence. The annotation of queries with classes may improve the retrieval, and so could the combination of the two proposed methods. Fact retrieval would benefit from the more precise connectedness approach whereas increasing the impact of the taxonomic approach would be preferable also in exploratory search system. Furthermore, the main ideas can be transferred to an adapted language or probabilistic retrieval model.

References

1. T. H. Cao, K. C. Le, and V. M. Ngo. Exploring combinations of ontological features and keywords for text retrieval. In T. B. Ho and Z.-H. Zhou, editors, *PRICAI*, volume 5351 of *LNCS*, pages 603–613. Springer, 2008.
2. P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):261–272, Feb 2007.
3. O. Egozi, E. Gabrilovich, and S. Markovitch. Concept-based feature generation and selection for information retrieval. In *AAAI*, volume 8, pages 1132–1137, 2008.
4. O. Egozi, S. Markovitch, and E. Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems*, 2011.

5. F. Giunchiglia, U. Kharkevich, and I. Zaihrayeu. Concept search. In *Proc. of the ESWC 2009*, pages 429–444, Berlin, Heidelberg, 2009. Springer-Verlag.
6. S. Harispe, S. Ranwez, S. Janaqi, and J. Montmain. Semantic measures for the comparison of units of language, concepts or entities from text and knowledge base analysis. *CoRR*, abs/1310.1285, 2013.
7. A. Hliaoutakis, G. Varelas, E. Voutsakis, E. G. M. Petrakis, and E. E. Milios. Information retrieval by semantic similarity. *Int. Journal on Semantic Web and Information Systems*, pages 55–73, 2006.
8. M. O. Jackson et al. *Social and economic networks*, volume 3. Princeton University Press, 2008.
9. J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. 1997.
10. D. Lin. Extracting collocations from text corpora. In *First Workshop on Computational Terminology*, pages 57–63, 1998.
11. G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.
12. V. M. Ngo and T. H. Cao. Ontology-based query expansion with latently related named entities for semantic text search. *Advances in Intelligent Information and Database Systems*, 2010.
13. P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of the 14th Int. Joint Conference on Artificial Intelligence*, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
14. N. A. L. Seco. Computational models of similarity in lexical ontologies. Technical report, Masters Thesis, 2005.
15. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.
16. T. Tietz, J. Waitelonis, J. Jäger, and H. Sack. Smart media navigator: Visualizing recommendations based on linked data. In *13th International Semantic Web Conference, Industry Track*, pages 48–51, 2014.
17. G. Tsatsaronis and V. Panagiotopoulou. A generalized vector space model for text retrieval based on semantic relatedness. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 70–78. Association for Computational Linguistics, 2009.
18. A. Tversky. Features of similarity. *Psychological Review*, 84(4):327, 1977.
19. R. Usbeck et al. GERBIL – general entity annotation benchmark framework. In *24th WWW conference*, 2015.
20. D. Vallet, M. Fernández, and P. Castells. An ontology-based information retrieval model. *The Semantic Web: Research and Applications*, 2005.
21. E. M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–180. ACM, 1993.
22. J. Waitelonis and H. Sack. Towards exploratory video search using linked data. *Multimedia Tools and Applications*, 59(2):645–672, 2012.
23. S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalized vector spaces model in information retrieval. In *Proc. of the 8th SIGIR Conf. on Research and Development in Information Retrieval*, pages 18–25, New York, NY, USA, 1985. ACM.
24. Z. Zhou, Y. Wang, and J. Gu. A new model of information content for semantic similarity in wordnet. In *Future Generation Communication and Networking Symposia, 2008. FGCNS '08. 2nd Int. Conf. on*, volume 3, pages 85–89, Dec 2008.