Research Article **Open Access**

José Devezas* and Sérgio Nunes

Hypergraph-of-entity

A unified representation model for the retrieval of text and knowledge

https://doi.org/10.1515/comp-2019-0006
Received December 5, 2018; accepted April 17, 2019

Abstract: Modern search is heavily powered by knowledge bases, but users still query using keywords or natural language. As search becomes increasingly dependent on the integration of text and knowledge, novel approaches for a unified representation of combined data present the opportunity to unlock new ranking strategies. We have previously proposed the graph-of-entity as a purely graph-based representation and retrieval model, however this model would scale poorly. We tackle the scalability issue by adapting the model so that it can be represented as a hypergraph. This enables a significant reduction of the number of (hyper)edges, in regard to the number of nodes, while nearly capturing the same amount of information. Moreover, such a higher-order data structure, presents the ability to capture richer types of relations, including *n*-ary connections such as synonymy, or subsumption. We present the hypergraph-of-entity as the next step in the graph-of-entity model, where we explore a ranking approach based on biased random walks. We evaluate the approaches using a subset of the INEX 2009 Wikipedia Collection. While performance is still below the state of the art, we were, in part, able to achieve a MAP score similar to TF-IDF and greatly improve indexing efficiency over the graph-of-entity.

Keywords: semantic search, hypergraph-based models, collection-based representation, text and knowledge unification

1 Introduction

Entity-oriented and semantic search are centered around the integration of unstructured data, in the form of text, and structured data, in the form of knowledge. We have frequently used data structures like graphs as a representation that promotes the integration of heterogeneous data. Hypergraphs take it even further, by providing a more expressive data structure that can, at the same time, capture both the relations and the intersections of nodes. We propose that hypergraphs should be used as an alternative data structure for indexing, not only because of their expressiveness — a document might be modeled as a hyperedge with terms and entities, potentially subsuming other relations between entities —, but also because they have the potential to scale better than a graph-based approach — in particular, relations like synonymy or co-occurrence can be modeled with only one hyperedge as opposed to creating a complete subgraph for all synonyms or co-occurring nodes. It is also clearer, from a semantics perspective, to model synonymy or co-occurrence as a single hyperedge. Even visually, a hypergraph can, through transparency, provide further insights regarding intersections and subsumptions [1, Figures 2 and 5].

Let us for instance assume a labeled hyperedge, *related_to*, which connects four entities mentioned in a document entitled "Cat": *Carnivora*, *Mammal*, *Felidae* and *Pet*. In subsumption theory, this hyperedge would represent an extension of *Carnivora*, *Mammal* and *Felidae*, a set of entities that could be a part of a hyperedge present for instance in a document entitled "Lion". While a lion is not a pet, it is still related to cat through the remaining three entities, so this information is useful for retrieval. While such example illustrates the potential of a hypergraph-based model, it only just scratches the surface. Using a hypergraph we can represent *n*-ary relations — linking more than two nodes (undirected hyperedge) or two sets of nodes (directed hyperedge) —, but also hierarchical relations (hyperedges contained within other hyperedges) and any partial combination of the two. This means we can, for instance, model synonyms as an undirected hyperedge (e.g, $\{result, consequence, effect, outcome\}$) and even introduce hypernyms/hyponyms as directed hyperedges (e.g., $\{cat, lion\} \rightarrow \{feline\}$).

To sustain our argument, let us consider an alternative approach based on a tree, which is a type of directed

***Corresponding Author: José Devezas:** INESC TEC and Faculty of Engineering, University of Porto, Portugal; E-mail: jld@fe.up.pt
Sérgio Nunes: INESC TEC and Faculty of Engineering, University of Porto, Portugal; E-mail: ssn@fe.up.pt

graph, to represent hierarchical relations. With a basic tree we lose the ability to simultaneously represent hierarchical and n-ary relations. We argue that a bipartite graph or an edge-labeled mixed graph would allow for the representation of both n-ary and hierarchical relations, but, while conceptually it would contain the same information as the hypergraph, it would also be harder to read and use. We wouldn't be able, for instance, to naturally identify intersections or subsumptions. Independently of whether or not we translate the hypergraph to an equivalent graph, at the very least the theoretical modeling power of the hypergraph is clear. Nonetheless, in practice there are also some advantages to using hypergraphs over graphs, for instance the fact that a single hyperedge can store all synonyms at once, requiring a single step to retrieve the synonyms for a single term — $O(|V|)$ for term nodes V. Conversely, the same operation on a graph would require as many steps as the number of synonyms for a single term — $O(|V| + |E|) = O(b^d)$, assuming a breadth-first search approach for term nodes V and synonym edges E or, equivalently, for outdegree b (the branching factor of the graph) and distance d (where d would be the same as the graph diameter).

Another advantage of hypergraphs includes the attempt to more closely model the human cognitive process. When we think, we inherently relate, generalize, particularize or overlap concepts. Most of us also translate natural language (sequences of terms) into concepts (entities), supporting the thought process on language. What we propose to do with the hypergraph-of-entity is to attempt to develop a kind of cognitive search engine (or at least the foundation for one). The way the (hyper)graph is traversed, including the selection of the point of origin, determines the kind of process over the "brain" of the engine. As a result, generalization becomes possible. With only slight adjustments to the search process, we can add support for multiple tasks from entity-oriented search. This includes ad hoc document and entity retrieval — the point of origin might be *term* nodes from a keyword query —, as well as related entity finding and entity list completion — the point of origin might be one or several example *entity* nodes; both tasks can also be considered a type of recommendation [2]. In order to avoid a combinatorial explosion while still taking advantage of structural features, we propose that we model each process using random walks over the hypergraph — each step is based on the random selection of an incident hyperedge and the subsequent random selection of one of its nodes. In this work, we focus on the task of ad hoc document retrieval, a process that we implement by modeling documents as hyperedges of terms and entities, and ranking *document* hyperedges through random walks. If we instead ranked *entity* nodes using the same strategy, we would have generalized the problem to ad hoc entity retrieval.

In Section 2, we clearly present the challenges and opportunities leading to this work, in particular distinguishing between considerations that impact the future of the model and the actual work presented in this paper. In Section 3, we present relevant work in entity-oriented and semantic search, as well as graph-based and hypergraph-based approaches. In Section 4, we describe the hypergraph-of-entity representation and retrieval model, introducing the base model, as well as three optional index features that can be combined as desired to extend the base model: synonyms, context and weights. We also describe the ranking approach, based on seed node selection and random walks. In Section 5, we present the test collection used for evaluating the ad hoc document retrieval performance, following with a study of rank stability, given our nondeterministic ranking approach, and the performance assessment for six models combining different index features of the hypergraph-of-entity. Finally, in Section 6, we conclude with some final remarks and directions for future work.

2 Problem statement

When answering a user's information need, entity-oriented search reconciles results from unstructured, semi-structured and structured data. This problem is frequently approached by establishing separate tasks, where the information need is solved as a combination of different subsystems. While each subsystem can use information from the other subsystems, they usually have their own central representation and retrieval model. For example, the inverted index is one of the main representation models in ad hoc document retrieval. And while structured information can be integrated into the inverted index to improve retrieval effectiveness, the rich and complex relations from knowledge bases are seldom transposed to the inverted index in an effective manner. For instance, related entities can be represented as text through a description or a profile. This way they can then be indexed in a field of the inverted index and contribute to the ranking function as any other field would. Another approach is to separately query the inverted index and the knowledge base and somehow combine document and entity weights. So, the approach is to either combine the output of two models or to translate one type of data to fit a chosen model and always work in that domain.

Both approaches represent a missed opportunity to cross-reference units of information from unstructured and structured sources. A similar case can be made for knowledge bases where indexes over triples can be queried through SPARQL, sometimes taking advantage of full-text search to filter fields. There is clearly an opportunity for a joint representation model, with ranking approaches that are generalizable to different units of information, and even to different tasks over those units. In this work, we focus on developing the groundwork for such a model.

2.1 Tackled problems

Although there is already work where unstructured and structured data are combined, models have been overly centered on one or the other type of data, frequently considering one of them as the external signal. It is in this lack of a balanced middle ground that we find the opportunity for a contribution. Our hypothesis is that, by proposing a representation and retrieval model where text and knowledge are seamlessly considered, we will be able to:

1. Jointly represent terms, entities and their relations in a single index;
2. Propose a generalized ranking function for multiple entity-oriented search tasks;
3. Improve overall retrieval effectiveness through the unification of information sources.

In the following sections, we further detail how we tackle items 1 and 2, proposing the hypergraph as the central data structure for representation, along with a general ranking strategy over that data structure.

2.1.1 Joint representation model

Graphs are a proven data structure for modeling heterogeneous data. They have been used to index text documents, as the underlying abstraction of hypertext and for the representation of knowledge bases. Their ubiquity across the relevant areas of entity-oriented search led us to propose a graph-based representation and retrieval model that combines terms, entities and their relations. In particular, we propose to use a weighted mixed hypergraph, since it can simultaneously and clearly express:

1. Undirected n-ary relations (e.g., bag-of-words, sentence, synonymy, context similarity);
2. Directed n-ary relations (e.g., a set of terms pointing to an entity, or a set of entities belonging to a category);
3. Hierarchical relations (e.g., subsumption);
4. Ontological relations (e.g., "Donald Duck" is both a *duck* and a *character* in a comic book);
5. Intersections (i.e., overlap is naturally captured by hyperedges and their shared nodes);
6. Uncertainty (e.g., knowing that there is 80% certainty that a set of terms are contextually similar can be translated into a weight in the respective *context* hyperedge);
7. User preference (e.g., a user rating "Back to the Future" with 5 stars can be translated into a higher weight for the corresponding *entity* node).

Moreover, hypergraphs enable us to decrease the number of (hyper)edges in relation to the number of nodes, by prioritizing n-ary relations over binary relations. This is an advantage in reducing the complexity and improving retrieval efficiency.

In this work, we propose a hypergraph-based representation model, called hypergraph-of-entity, for the joint indexing of terms, entities and their relations. We explore most of the items that we previously listed, for expressing different relations in our model. In particular, we do not explore hierarchical relations, and we capture entity co-occurrence rather than explicit ontological relations, introducing uncertainty only on the weighted version of the model. This is further detailed in Section 4.1.

2.1.2 Generalized ranking function

Regarding the retrieval model, we propose a generalized ranking function over the hypergraph-of-entity. One of our ongoing goals is to design a function that can be used independently of the unity of information, as well as for multiple different tasks. We suggest this should be done by controlling:

- Input and output, e.g.:
  - Input *term* nodes to output a ranking of *document* hyperedges;
  - Input *entity* nodes to output a ranking of other *entity* nodes.
- Parameter configuration, e.g.:
  - Longer traversals will be more exploratory;
  - Shorter traversals will be more precise or on-topic.

In particular, the approach we propose is based on:

1. Finding a representation for the query in the hypergraph-of-entity;
2. Ranking nodes and hyperedges based on traversals around those representations;

3. Collecting only the relevant nodes or hyperedges to present to the user.

In this work, we propose a generalized ranking function, which, at this stage, we only evaluate for the ad hoc document retrieval task. The model, however, is designed to easily support tasks like ad hoc entity retrieval, related entity finding or entity list completion, which we plan to assess in the future. The following section further details each retrieval task and presents material to support our vision for unification over (hyper)graph-based data structures.

3 Reference work

In this section, we introduce entity-oriented and semantic search. We first cover query-dependent and query-independent evidence, from classical models to entity-oriented models. We then present an overview on the main retrieval tasks in entity-oriented search, that are the focus of the generalized model we propose in this work. Next, we discuss the opportunity for unified models, also commenting on the need for combined data with associated relevance judgments over multiple tasks. Finally, we present graph-based and hypergraph-based models and their applications in information retrieval and the representation of documents. We close the section with a discussion on the usefulness of hypergraphs to model cognitive functions in the brain, which we associate with a generalized model for entity-oriented search.

3.1 Entity-oriented and semantic search

Entity-oriented and semantic search branches into multiple tasks, where each task takes one or multiple types of queries — unstructured, semi-structured, structured — and returns one or several types of results — documents, entities (based on relevance, relatedness, etc.). For instance, semantic information can be used to improve ad hoc document retrieval, (where queries are keywords and results are documents, but knowledge is used to inform retrieval), or it can be used for entity list completion (where queries can be keywords or a selection of entities, either way representing example entities, and results are entities). With over 80% of queries mentioning entities [3], entity-oriented search has become a relevant problem within information retrieval. While the inverted index has remained central in tackling this challenge, in the last few years there has also been work in graph-based approaches for information retrieval [4, 5], and a growing exploration of unified models [6, 7].

Hybrid collections containing text, entities and their relations are essential to the study of joint representation models, in particular when accompanied by relevance judgments for multiple tasks. These hybrid collections are also called combined data. Bast et al. [8, Definition 2.3] have defined combined data based on two principles: *link* — text annotated with (or linked to) entities from a knowledge base (e.g., through named entity recognition and disambiguation, or based on hyperlinks); and *mult* — combined knowledge bases with different naming schemes (e.g., through automatic ontology matching, or based on a manually curated high-level ontology).

In this work, we define a joint representation model that works for combined data. Moreover, we use it as an index data structure for entity-oriented search, designed to support multiple tasks in this domain. The approach we propose eliminates the need for multiple models or even the need to change the model depending on the task. We focus on assessing ad hoc document retrieval, proposing a hypergraph-based model to extend text-based retrieval with information from entities and their relations. In order to assess effectiveness, we take advantage of the INEX 2009 Wikipedia Collection [9], which includes semi-structured data from Wikipedia (text from Wikipedia articles, annotated with links to related Wikipedia articles, which we use as a knowledge base).

3.1.1 Query-dependent and query-independent evidence

The goal of search engines is to help users solve their information needs, which are usually expressed as keyword or natural language queries. Documents or entities are then weighted according to the query and a list of ranked results is provided to the user. Many of the classic weighting functions used in traditional search engines can also be applied to entity-oriented search.

While in search engines the relevance of a document is definitely query-dependent, query-independent evidence can still be used to better discriminate documents. Notable query-dependent approaches include TF-IDF [10, 11], BM25 [12], language models [13] or divergence from randomness [14]. In Section 3.1.2 we show an example of how the task of ad hoc entity retrieval can be mapped to the task of ad hoc document retrieval, as a way to reuse these existing ranking functions. Query-independent approaches, on the other hand, include document priors like document

length, number of incoming links or URL depth [15, 16], as well as the well-known PageRank algorithm [17], along with social signals based on the number of likes, shares or bookmarks in different social media platforms [18], or even based on specific Facebook emotions [19]. Other work has also used documents as indicators of expertise [20], slightly reversing the roles (i.e., documents as entity priors; see also Fang and Si [21]). Additionally, there has been work showing that closely linked documents usually cover similar topics [22]. This is also know as the cluster hypothesis, which has been shown to be true for web-based collections [23]. It is a necessary condition to be able to model relevance using proximity-based traversals over graphs or hypergraphs. The work we describe here takes advantage of links in semi-structured data from Wikipedia, modeling the collection as a hypergraph, and retrieval as short random walks in that hypergraph.

One of the approaches for ranking over (hyper)graphs is to use random walks based strategies. PageRank is an example of a query-independent feature that can model the influence of a page in the web graph. Over time, several PageRank variants have been proposed and, in particular for entity-oriented search, there are some interesting applications worth mentioning. These include ReConRank [24], ObjectRank [25], PopRank [26], HubRank [27] and DatasetRank [28]. Overall, these algorithms aim at providing better link analysis for the semantic web, by integrating information from the web graph (or some other context-establishing element like "dataset") with information from a knowledge graph (links between objects or entities, usually with different weights for different types of relations). The work we present here shares some of the ideas introduced by these approaches, but proposes a hypergraph-based model, where context is established by hyperedges and ranking is done through simulated random walks.

3.1.2 Retrieval tasks

In order to propose a generalized model, we must first identify the commonalities between the elements we are attempting to model. In entity-oriented search, this means identifying which elements we should represent, which elements should be used to query and which elements should be ranked and retrieved. In this section, we provide an overview on the retrieval tasks from entity-oriented search, in particular describing some of the representation and retrieval models used to tackle each task.

Ad hoc document retrieval

It is frequent for modern search engines to return a rich assortment of results, including documents, entity lists and information cards, direct answers, etc. Ad hoc document retrieval has, however, remained a central task in the area, improving its effectiveness through semantics. Entities and their relations can be harnessed to improve the traditional process of document retrieval by furthering informing retrieval. Raviv et al. [29] have proposed such a method, where they have enhanced document retrieval using entity-based language models. Interestingly, their model accounted for the uncertainty that is inherent to entity linking, which we are also interested in exploring as part of the general ranking approach over the hypergraph-of-entity. They also explored the balance between term-based and entity-based information. In particular, they experimented with cluster-based document retrieval, while testing several combinations of term-based and entity-based language models to induce clusters, as well as document-query similarities.

Ad hoc entity retrieval

Ad hoc entity retrieval is one of the fundamental tasks in entity-oriented search. It consists of taking a keyword query, potentially containing natural language segments, and transforming it into a ranked list of entities — sometimes also called objects or "things" [30]. The challenge is combining associated textual passages with underlying knowledge that accompanies the mentioned entities, not only to improve effectiveness, but also to provide novel ways of harnessing all available information. This is frequently achieved through virtual documents [3], learning to rank [31–33] or the integration of signals separately obtained from an inverted index and a triplestore [34]. Our hypothesis is that we are missing out on the opportunity to follow the leads across the boundaries of text, through the space of knowledge and back, as needed, by using separate models that only coalesce during ranking.

Related entity finding

Another central task in entity-oriented search is related entity finding. Given a source entity, a target type and the nature of the relation (e.g., `[ bands like Slayer ]`, where the source entity is given by *Slayer*, the target type by *bands* and the relation by *like*), find other entities of the given target type that respect the specified relation (e.g., *Anthrax*, *Metallica* and *Kreator* are all *bands* that share a common genre or are *like Slayer*). Cao et al. [35] proposed a bipartite graph based method for related entity finding, built from

the co-occurrence of entities in unstructured and structured lists. They first identified candidate entities of the given target type, calculating an initial relevance score. At that point, some related but unpopular entities would rank lower than expected. In order to solve this issue, they used the bipartite graph containing two disjoint sets, one for candidate entities (initially scored) and another one for the instances where candidate entities occurred (the lists). This means that, whenever a candidate entity occurred in a given instance, the respective nodes would be linked. In order to compute the final relevance score, they used a process analogous to heat diffusion over the graph, in order to propagate the initial relevance scores until convergence. This process relied on the idea that entities similar to relevant entities are also relevant (according to the cluster hypothesis for entity-oriented search [36]). This resulted in the boosting of unpopular but related entities using list co-occurrence as an indicator of similarity. This is an interesting approach, regarding what we propose in this work, for two reasons: first, there is a relation between bipartite graphs and hypergraphs, since an instance node could be represented as a hyperedge of candidate entity nodes instead; secondly, the authors propose an alternate method to random walks for propagating weights over the graph (heat diffusion).

Entity list completion

Another important task in entity-oriented search is entity list completion. This task is similar to related entity finding, but also takes into consideration a given set of example entities that serve as relevance feedback. The goal is to rank and retrieve other similar entities (e.g., given *Slayer* as the source entity, *bands* as the target type and *like* as the relation, as well as *Anthrax*, *Sepultura* and *Metallica* as the examples, the list could be completed with entities like *Kreator*, *Megadeth* and *Lamb of God*, which are thrash metal bands like the source and example entities). Bron et al. [37] compared text-based and structure-based approaches for entity list completion, finding that both approaches were effective, despite returning different results. This led them to experiment with linear combinations of both approaches, as well as a method that switched between either approach depending on the predicted effectiveness (using example entities as relevance judgment). Their experiments have shown that combining text and knowledge outperformed either one of the approaches when independently used. One question that remains is whether highly hybrid methodologies, that indiscriminately take advantage of either terms or entities and their relations, are able to perform better.

Fang and Si [21] have proposed two unified probabilistic models for related entity finding, one of which they also applied to entity list completion (ELC). While, for this second task, they ignored the example entities provided in the topics, they were still able to reach the best performance, according to MAP, for the ELC task in TREC 2010 Entity track. Both models considered probabilistic components for candidate entity type, expected entity type and type matching. Model A contained probabilistic components for entity relevance, as well as for source and target entity co-occurrence, while Model B contained a probabilistic component for entity relevance without considering the source entity, as well as an entity prior component. The difference was in Model B ignoring the source entity. Experiments, however, showed a better performance for Model A, when compared to Model B, supporting the importance of the source entity in the modeling process. The good overall performance of such a holistic probability framework illustrates the importance of a unified model that is able to capture the complex relations between the units of information.

3.1.3 Why a unified model?

Pound et al. [38] have proposed five query categories for entity-oriented search — entity query, type query, attribute query, relation query, and other keyword query. These, in many cases, can be mapped into specific tasks of entity-oriented search. For instance, an entity query could be solved through ad hoc entity retrieval, while a type or relation query might be solved through related entity finding or entity list completion. Devezas et al. [39] have also shown that it is possible to use a graph to implement multiple recommendation tasks that are not unlike some of the tasks we have presented in this section. This adds to the evidence that a joint representation and retrieval model might be viable and generalization might be possible.

Furthermore, when using combined data for multiple different retrieval tasks, the question of generalization in information retrieval intensifies. Is it possible to devise a representation model that is capable of integrating heterogeneous information (or, more simply, text, entities and their relations), as well as an associated retrieval model that is able to, with the configuration of some parameters (or through sequential output), return each of the elements of the expected rich results? We already know that learning to rank [31, 33] is a good method to integrate features from potentially heterogeneous data, but we would need to train a model for each of the tasks (e.g., one to predict document relevance based on the query, and an-

other one to predict related entities based on one or several of the mentioned entities). Training separate models is completely acceptable and it might even improve modularity from an engineering perspective, however the question of generalization remains. Is it possible? Moreover, if it is possible, what type of gain or unpredicted consequences would arise from such a unified model? Will it result in improved retrieval effectiveness or decreased performance? We explore this idea, beginning with the proposal of a generalized hypergraph-based model, designed to support entity-oriented search tasks, such as ad hoc document retrieval when enhanced with structured knowledge.

3.1.4 Test collections

We built our evaluation framework on top of the INEX 2009 Wikipedia Collection [9]. This collection, which is further described in Section 5.1, provides a way to assess the performance of ad hoc document retrieval, through the INEX Ad Hoc track [40]. It also provides relevance judgments from INEX XML Entity Ranking track [41], which can be used for assessing entity ranking, as well as entity list completion, in our future experiments. There are, however, other test collections that can be considered for the evaluation of generalized retrieval models in entity-oriented search. These include ClueWeb09[1] with relevance judgments from TREC Web track [42] and TREC Entity track [43]; Sindice-2011 Dataset [44] with relevance judgments from TREC Entity track; and TREC Washington Post Corpus[2] with relevance judgments from TREC Common Core track[3] and TREC News track[4]. However, for the reasons we enumerate next, we didn't use any of these collections for evaluation. ClueWeb09 is a large collection, with web-scale challenges that are out of the scope of this work. Despite its recognizable relevance in early entity-oriented search research, Sindice-2011 Dataset is no longer easily available (we could not find it on the web). Furthermore, no associated relevance judgments for ad hoc document retrieval were found. TREC Washington Post Corpus is a fairly recent dataset, which had not yet been released at the time of preparing our experimental framework. However, it is an appropriate and easily available test collection that we will consider in the future.

1 http://lemurproject.org/clueweb09/
2 https://trec.nist.gov/data/wapost/
3 https://trec-core.github.io/2018/
4 http://trec-news.org/

3.2 Graph-based models

Graphs are particularly good for integrating data and, while they are inherently used to model knowledge (e.g., through ontologies), they can also be used to represent text [4, 5, 45, 46] and their relations [4, 5, 45, 47, 48].

Blanco and Lioma [4] have proposed a graph of terms able to capture context by linking co-occurring terms within a window of size N, using either undirected edges, or directed edges to express grammatical constraints. Rousseau and Vazirgiannis [5] have also explored this idea by proposing that directed edges should instead be used to link each term at the beginning of the window to its following terms within that window, thus capturing term dependency instead of grammatical constraints. Both approaches defined a graph of terms based on co-occurrence, but they did not include any entity-based information. More recently, Zhu et al. [49] proposed a natural language interface to a graph-based bibliographic information retrieval system. Through named entity recognition and dependency parsing, they were able to generate a graph query that was capable of correctly interpreting 39 out of 40 natural language queries of varied complexities. Despite some domain-dependent limitations, introduced for higher performance, they presented an interesting result: graphs have the potential to significantly aid in query interpretation, thus improving overall retrieval effectiveness.

On the other side of the spectrum, focusing on knowledge instead of text, Blanco et al. [50] have explored the problems of effectiveness and efficiency for ad hoc entity retrieval over RDF data. Their ranking approach was based on BM25F, experimenting with three representation models: (*i*) an horizontal index, where fields *token*, *property* and *subject* respectively stored terms, RDF properties and terms from the subject URI; (*ii*) a vertical index, where each field represented a separate RDF property containing terms from the respective literals; and (*iii*) a reduced version of the vertical index where fields represented *important*, *neutral* and *unimportant* values according to their weight. While this approach enabled search over a knowledge base and, to some degree, integrated text and knowledge, it still missed on the opportunity to capture the implicit relations that we so naturally use as part of our cognitive process. We can easily derive new knowledge from text, alternating between potentially incomplete fragments of text and knowledge and following them as leads to our destination.

In this work, not only we attempt to integrate text and knowledge in a single representation model, but we refrain from prematurely moving into the inverted index based

on the extraction of features from the (hyper)graph-based representation. Despite obvious efficiency issues, we are, at this stage, purely focused on exploring the potential of the (hyper)graph as the sole data structure behind the retrieval process. Hypergraphs, in particular, have the potential to take even further what graphs already provide by modeling not only binary relations, but also *n*-ary and hierarchical relations, while capturing intersections between groups of nodes.

3.3 Hypergraph-based models

While hypergraphs have been previously used in information retrieval, they still don't play a major role in well-known tasks, despite their potential. The most notable work we found was the query hypergraph proposed by Bendersky and Croft [51], where vertices represent concepts from the query, and edges represent the dependencies between subsets of those vertices. Based on the idea of a factor graph (a bipartite graph or a kind of hypergraph), they proposed a ranking function to obtain a relevance score of a document given a query, based on local and global factors, which worked as hyperedge weights. Their hypergraph was able to represent *higher-order term dependencies*, therefore modeling *dependencies between term dependencies*, which neither the Markov random field model or the linear discriminant model were able to do, despite similarities within the ranking functions. They defined two types of hyperedges: local, between individual concepts and the document, and global, between the entire set of concepts and the document. Their methodical approach can be regarded as a fundamental step in supporting hypergraph-based work in information retrieval. In this work, we explore multiple types of relations between concepts, however our concepts consist not only of terms, but also of entities, and we mention weights as opposed to factors.

Hypergraphs have also been recently used for summarization [52], as an XML alternative for the semi-structured representation of text as a graph [53] or to model folksonomies, promoting the serendipitous discovery of new content [54]. Even in 1981, in the area of social network analysis, Seidman [55] had noticed the inability of anthropologists and sociologists to study social networks based only on dyadic relationships, proposing hypergraphs as a way of better modeling non-dyadic relationships, such as group membership. Moreover, hypergraphs have already been particularly useful in music recommendation [6, 56–58] through unified approaches for modeling heterogeneous data or through the use of random walks.

In his last lecture, von Neumann [59] discussed how the brain can be viewed as a computing machine, thus reinforcing the relevance for cooperation between computer science and neuroscience. Interestingly, there is evidence in cognitive science of the relevance of hypergraphs in modeling functional connectivity in the brain [60–63], as well as learning and memory [64]. In particular, the work by Gu et al. [63] in neuroscience has led to three hyperedge archetypes — stars, bridges and clusters —, two of which we also use in this work — clusters of terms and entities as documents, and bridges established by synonyms and context. Hypergraphs have also been proposed as a model for the creation of artificial general intelligence [65], which would be fundamental for a cognitive search engine. Assuming that we would be able to effectively represent text and knowledge using a hypergraph, then we might be able to take advantage of both set theory, using metrics like the Jaccard index to measure similarities, or random walks in hypergraphs [66], where we might use hyperedge weights, but also node weights to control the traversal. These are ideas that we explore in this work, aiming at understanding if hypergraphs have the potential to improve retrieval effectiveness, assuming text and knowledge as heterogeneous but strongly related data, that power the process of entity-oriented search.

We have seen that graphs can be used to represent both unstructured text (e.g., graph-of-word) and structured knowledge (e.g., DBpedia). Hypergraphs can go even further, capturing for instance synonyms as a single hyperedge. In previous work [67], we have already explored the unification of terms, entities and their relations as a graph, proposing the graph-of-entity as a representation model for entity-oriented search. In our experiments, we encountered both indexing and retrieval challenges, in particular regarding efficiency. With the hypergraph-based model we describe here, some of those issues were mitigated. In particular, it was possible to greatly reduce the number of edges in the original graph-of-entity. With only slight modifications, we were able to group a larger number of related nodes through hyperedges. This is yet another advantage of hypergraph-based representations, that is further detailed in Section 5.4.1.

4 Hypergraph-of-entity

Ad hoc document retrieval is traditionally a text retrieval task. Semantic search, however, frequently takes advantage of annotated collections, where entities are recognized and linked to external knowledge bases to improve

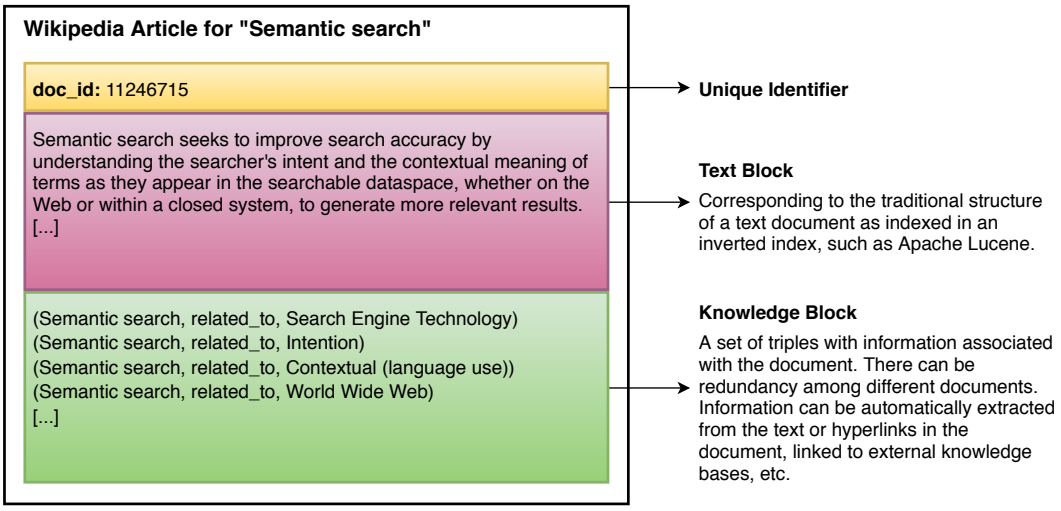


Figure 1: Extended document definition for combined data. Example based on the Wikipedia article about "Semantic search".

document retrieval. In this work, we assume that, like entity annotations, relevant relations are also a part of the annotated document, extending it. Given a document containing a text block of unstructured data, as well as a knowledge block of structured information (i.e., entities and relations that are relevant to the document), our goal is to propose a joint representation model able to provide seamless integration, as well as support for entity-oriented search tasks, from ad hoc document retrieval to related entity finding. A regular document usually contains multiple text fields (e.g., *title*, *content*, etc.), which corresponds to the text block in the extended document. However, we also include a knowledge block, in the form of triples, that are usually available as structured data in the original document. The knowledge block can be directly extracted from a semi-structured document (e.g., building triples based on links to other documents), but it might also be obtained from an information extraction pipeline. There is no restriction about the source of the knowledge block, except that it should represent a set of triples related to the document. For example, the triples might represent co-occurring entities in a sentence or paragraph, or statements obtained from a dependency parser, or they could represent external knowledge about identified entities, from an external knowledge base.

Figure 1 illustrates such an extended document based on the Wikipedia article for "Semantic search", and it includes a unique identifier, the text block describing the entity, and the knowledge block containing triples based on hyperlinks (i.e., using Wikipedia as the knowledge base). We propose that a hypergraph would be the ideal data structure to represent a collection of extended documents, effectively capturing the dependencies and higher-order dependencies between terms and entities in relation to the documents. Take for example a *document* hyperedge created to associate all the elements within a document, including its terms and entities. Through higher-order dependencies we are, for instance, able to capture subsumption, where documents subsume (i.e., are more general than) relations between entities — we might interpret it as "document d_1 explains the relations between entities e_1, e_2 and e_3". The hypergraph-based model, detailed in Section 4.1, is able to capture multiple levels of information about the text, the entities and their relations, providing a more unified and insightful view over all available information. Although in this contribution we do not explicitly assess the impact of subsumption or hierarchical relations, but only of n-ary relations based on synonymy and context, we do highlight the ability for the model to capture such complex relations. Moreover, regarding ranking, document relevance scoring is based on biased random walks over the hypergraph, departing from a set of term and entity nodes that represent the query. This is a ranking approach that closely depends on the structure of the hypergraph, making it easier to track the impact that changes to the representation have in retrieval performance. The retrieval model is detailed in Section 4.2.

4.1 Representation

In this section, we introduce the variations of the hypergraph-of-entity representation model. This includes the base model, as well as multiple extensions based on synonyms, context and weights.

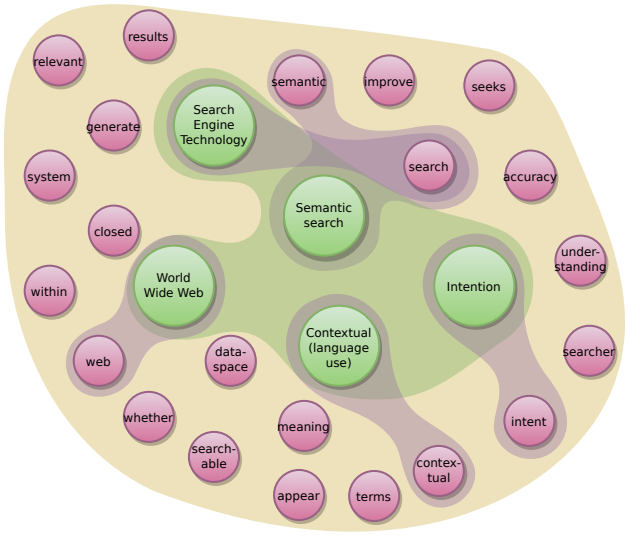


Figure 2: Hypergraph-of-entity base model, representing the first sentence of the Wikipedia article for "Semantic Search".

4.1.1 Base model

The hypergraph-of-entity is, in many ways, a simplification of the graph-of-entity [67]. In the graph-of-entity, the sequence of terms in a document was captured through term–term edges with a *doc_id* attribute, while in the hypergraph-of-entity we discarded term dependency in order to be able to model the terms within a document as a single hyperedge. This model is analogous to the bag-of-words, in the sense that term dependency is not captured by hyperedges (sets of nodes). Besides this major difference (one hyperedge per document), there are three other notable differences between the hypergraph-of-entity and the graph-of-entity: (*i*) each document hyperedge also contains nodes for entities mentioned within the document; (*ii*) sets of entities can be linked through a *related_to* hyperedge; and (*iii*) sets of terms can be related to an entity through a *contained_in* hyperedge. We use a mixed hypergraph to represent a collection of documents. This means that hyperedges can be directed — from a set of terms to an entity (*contained_in*) — or undirected — sets of terms and entities (*document*), and sets of related entities (*related_to*).

In undirected hypergraphs, a set of nodes is a hyperedge. In directed hypergraphs, a hyperedge (or hyperarc) contains two sets of nodes — the set of source nodes is called tail, while the set of target nodes is called head. In the hypergraph-of-entity, we always have tail sets with cardinality one (for directed *contained_in* hyperedges) — this characteristic might be useful for defining a tensor representation of the hypergraph. Figure 2 provides a basic illustration of this model, without capturing hyperedge direction. In the figure, pink nodes represent terms and green nodes represent entities. All term and entity nodes are linked by a yellow undirected hyperedge that represents the document as the set of its terms and entities. Entity nodes are linked by green undirected hyperedges, when the entities are related (e.g., through a property in an ontology). Sets of term nodes are linked to an entity by a pink directed hyperedge, whenever the terms are a good representation of the entity (e.g., through substring matching).

In particular, Figure 2 represents a single document based on the first sentence of the "Semantic Search" Wikipedia article:

> *Semantic search seeks to improve search [Search Engine Technology] accuracy by understanding the searcher's intent [Intention] and the contextual [Contextual (language use)] meaning of terms as they appear in the searchable dataspace, whether on the Web [World Wide Web] or within a closed system, to generate more relevant results.*

Underlined terms within the text block represent links to other Wikipedia entities (shown in square brackets). This establishes the knowledge block, already depicted in Figure 1. Each term obtained from the tokenization of the text block is represented only once within a document hyperedge, regardless of its frequency within that document. The same happens when multiple links to the same entity are found — the entity is always represented by the same, unique node. A similar structure is found in the documents of INEX 2009 Wikipedia Collection, which we use to evaluate our model (cf. Section 5).

In order to better visualize the differences between this representation and the graph-of-entity, we recommend comparing Figure 2 with the right side of Fig. 1 from Devezas and Nunes [68], which indexes the same document we illustrate here. For that particular instance of the graph-of-entity, we had 22 *term* nodes and 5 *entity* nodes (the same number as the hypergraph-of-entity), but we also had 32 edges as opposed to only 7 hyperedges in the model we propose here. Such a significant edge reduction was in part possible because of the loss of term dependencies, when switching to the hypergraph-of-entity from the graph-of-entity.

4.1.2 Extensions

We provide three types of extensions to the base model — synonyms, context and weights —, which we can combine and reorder in any way we want. Extending the base model with synonymy and contextual relations provides a

kind of "organic" query expansion. We usually depend on query expansion to retrieve previously unreachable documents that did not match the user's vocabulary. With the hypergraph-of-entity, this becomes an inherent part of the ranking process, as it simply requires the addition of new hyperedges linking to related terms. On the other hand, introducing node weights enables term and entity boosting, and introducing hyperedge weights enables document boosting and the assignment of certainty to the information represented by the hyperedge, thus constricting the flow of random walks and directing the walker through the most probable paths. In this section, we provide further details on how synonyms, context and weights were obtained and added to the hypergraph-of-entity.

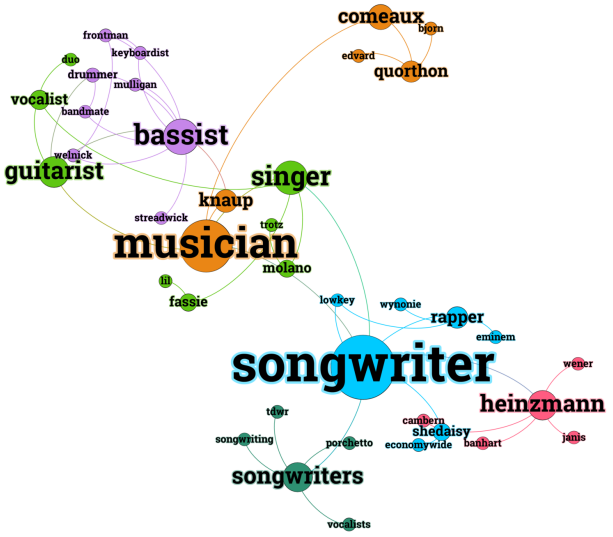


Figure 3: Word2Vec SimNet: "musician" ego network, with a depth of three. Nodes size is proportional to the betweenness centrality and colors identify clusters of densely connected terms.

4.1.2.1 Synonyms

We used WordNet 3.0, through JWI, the MIT Java Wordnet Interface[5], to obtain the *synset* for the first sense of each term (i.e., the sense that is more frequently used), assuming that the term is a noun. We integrated synonyms into the hypergraph-of-entity by adding missing term nodes (i.e., that were not originally a part of the collection's vocabulary) and linking all terms from the *synset* using a *synonym* hyperedge. For example, if a document contained the term "results", we would search WordNet as follows:

```
$ wn results -synsn

Synonyms/Hypernyms (Ordered by Estimated Frequency) of
noun result

4 senses of result

Sense 1
consequence, effect, outcome, result, event, issue, upshot
       => phenomenon

Sense 2
solution, answer, result, resolution, solvent
       => statement

Sense 3
result, resultant, final result, outcome, termination
       => ending, conclusion, finish

Sense 4
resultant role, result
       => semantic role, participant role
```

For this particular case, we would obtain four senses. We would then take the synonyms (i.e., the *synset*) from Sense 1 and link all the terms using a *synonym* hyperedge consisting of the following set of terms: "results" (the original term), "consequence", "effect", "outcome", "result", "event", "issue" and "upshot". At the same time, we stored information about the number of senses for each term, as it is useful to compute the weight of the *synonym* hyperedges.

4.1.2.2 Context

Two terms were considered contextually similar whenever they were frequently surrounded by similar terms. We used word2vec [69] word embeddings to establish context, based on the implementation provided by Gensim[6]. The model can either be trained with the same collection that is being indexed, or use an external text collection that might be more relevant to impose context within the given domain. Several hyperparameters can be tuned to control word2vec. We extracted word embeddings of size 100, considering moving windows with 5 words and discarding words with a frequency below 2 in the collection. Once we extracted the embeddings for all terms in the collection, we used the cosine similarity to find the k-nearest neighbors, with $k = 2$, building a similarity network where an edge was created between a term and each of its nearest

5 https://projects.csail.mit.edu/jwi/

6 https://radimrehurek.com/gensim/models/word2vec.html

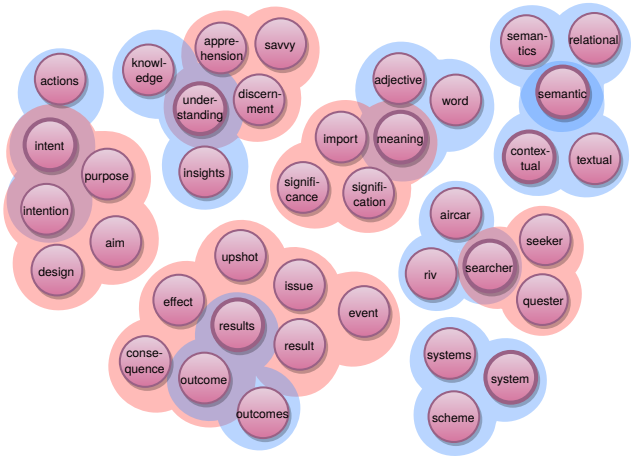


Figure 4: Hypergraph-of-entity model partial view, showing some of the new *synonym* (red) and *context* (blue) hyperedges. Term nodes from the original document are displayed with a stronger border stroke.

neighbors, but only when the similarity was above a given threshold (we used 0.5). Throughout this article, we call this the Word2Vec SimNet. Figure 3 shows the neighborhood of "musician" (its context), up to a maximum of three nodes in distance, in the Word2Vec SimNet for the INEX 2009 subset (see Section 5.1). As we can see, even if a query for "guitarist" or "bassist" is issued, documents containing only "musician" can also be considered, although expanding from "guitarist" should result in a higher weight to documents containing "musician" than expanding from "bassist", since "musician" is adjacent to "guitarist", but "bassist" can only reach "musician" through "guitarist". This is the kind of rationale that a graph-based design supports, simultaneously allowing for a better explanation and the promotion of transparency. We integrated this graph-based information into the hypergraph by creating an undirected *context* hyperedge, linking each term to all of its contextually adjacent terms. Were the user to require an explanation as to why a particular ranking was provided for a given query, we would be able to list the paths traversed from the seed nodes representing the query. We could either do it exhaustively (i.e., list all paths), or based on descriptive statistics, like the number of paths leading to ranked nodes, along with a few examples. Either way, graph-based or hypergraph-based models are easily traceable.

Figure 4 illustrates the hypergraph-of-entity revision, showing only *synonym* and *context* hyperedges, both examples of *n*-ary relations between multiple term nodes. We also added any missing term nodes that were external to the document, but present in the list of synonyms or contextually similar terms (in the figure, we only included some of the original terms to illustrate the different patterns). All nodes within blue *context* hyperedges were already a natural part of the hypergraph (i.e., contained in the original collection), since word2vec was trained with the same collection. However, synonyms might be external to the collection, therefore resulting in the addition of new term nodes that are not a part of any document. As we have seen before, both synonyms and contextually similar terms establish bridges between potentially disconnected, but related, documents, increasing the chances of improving recall over the base model. In the figure, nodes that are a part of the document are visually identified by a stronger border. Most of the document term nodes are not synonyms or contextually similar to one another. However, the terms "semantic" and "contextual" are both connected, since "contextual" is one of the top-2 most similar terms to "semantic", according to their word embeddings. Other interesting subhypergraphs include for instance the neighborhood of term "results", that contains appropriate synonyms like "consequence", "result" (the singular) or "effect", but also less clear synonyms like "issue" or "upshot"; contextually, however, we are able to reach both "outcome" and "outcomes", with "outcome" already covered by the synonyms (but not its plural), an indicator that relevant related terms might only be reachable through context.

4.1.2.3 Weights

By default, all nodes and hyperedges were unweighted. As another extension to the index, we assigned probabilistic weights to nodes and hyperedges. We did this for two main reasons. First, not all terms or entities (our nodes) are equally relevant, from a query-independent perspective; the same happens for related entities, contextual terms, or synonyms that depend on word sense (our hyperedges). Secondly, assigning weights might serve as a base for pruning in the future, which we predict might improve overall performance. Regarding effectiveness, constricting available paths is a way of increasing focus in the model and thus of guiding random walks. Regarding efficiency, a lower number of nodes and hyperedges result in a lesser amount of used memory, but also in a faster convergence of random walk visit probability, thus requiring less CPU cycles to reach an optimal result. On the other side, the non-uniform random selection of a node or hyperedge during random walks is more expensive than selecting an incident node or hyperedge uniformly at random, which means this requires experimentation.

The aim of the weights assigned to nodes and hyperedges was to provide discriminative power, thus requir-

Table 1: Hypergraph-of-entity weighting functions.

(a) Nodes.

Node	Weight	Description
term	$2S\left(\alpha\frac{N-n_t}{n_t}\right)-1$	We used a variation of the IDF, with a tunable $\alpha < 1$ parameter to control how fast the function decreases. - S is the sigmoid function - N is the number of documents in the collection - n_t is the number of documents where a given term t occurs. - We used $\alpha = N^{-0.75}$.
entity	Same as *term*.	In the future, we will experiment with different values of α for terms and entities, in particular alternative exponents to -0.75.

(b) Hyperedges.

Hyperedge	Weight	Description
document	0.5	Linking a term or entity simply through document co-occurrence is weak, so we use a constant weight lower than one.
related_to	$\frac{1}{\lvert E\rvert}\sum_{v\in E}\frac{\lvert\{u\in F \mid F\in\mathfrak{E}\setminus\{E\}\wedge v\in F\}\rvert}{\lvert E\rvert}$	For each entity within the hyperedge, we calculate the fraction of reachable other entities and average all results. - $\mathfrak{E}$ is the set of all *related_to* hyperedges. - $E \in \mathfrak{E}$ is the specific *related_to* hyperedge, for which we are calculating the weight.
contained_in	$\frac{1}{\lvert terms\rvert}$	We want links with fewer *terms* to be more frequently followed, since certainty that any term within the hyperedge leads to the entity is higher.
synonym	$\frac{1}{\lvert senses\rvert}$	The higher the number of possible senses, the less certain we are about the hyperedge, since we use the first (and most probable) sense according to WordNet.
context	$\frac{1}{\lvert terms\rvert}\sum_i\frac{sim(term_k,term_i)\times min_{sim}}{min_{sim}}$	A context is only as good as the average of all similarities between the original $term_k$ and all other $term_i$. We normalize the weight taking into account the threshold used to create the Word2Vec SimNet.

ing uniform distributions with well dispersed values. In this work, we provide an initial approach to weighting in the hypergraph. Table 1 provides an overview of the probabilistic weighting functions that we propose, based on the characteristics of each individual node and hyperedge type. For this first experiment with a weighted version of the hypergraph-of-entity, we selected weighting functions that we could compute exclusively using information internal to the model. In an attempt to ensure the generalization of the model, we also restricted the weights to probabilities, in order to facilitate the eventual integration of elements from probabilistic information retrieval or language models in the future.

In particular, for the weighting of terms and entities, we used the probabilistic IDF [70], but replaced the log function with the sigmoid function, to ensure that IDF would always range between zero and one. In the sigmoid IDF we provide a parameter α that controls the function's decrease speed. We manually experimented with multiple values for α, finding that the behavior would significantly change for collections of a different dimension. We, therefore, introduced a dependence on a fraction of the collec-

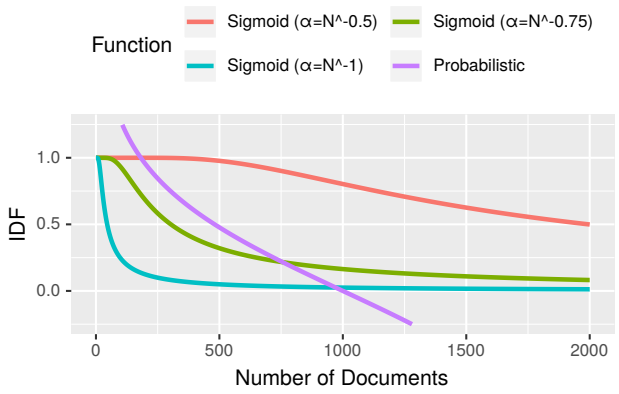


Figure 5: Selecting α for sigmoid IDF, when compared to the probabilistic IDF.

tion size N. Figure 5 illustrates the behavior of the probabilistic IDF when compared to sigmoid IDF for base N and exponents -0.5, -0.75 and -1. As we can see, using an exponent of -1 results in IDF values always being above 0.5 and a slow decrease behavior. On the other hand, using an exponent of -0.5 will result in a fast decrease with a large fraction of the collection with an IDF closer to zero. Finally, using -0.75 results in a decrease speed that is closer to the behavior of the probabilistic IDF assigning a more diverse range of values to different documents in the collection. While we did not specifically tune α to the best approximation to the probabilistic IDF, the value that we selected provides a good enough discriminative power.

4.2 Retrieval

We propose a ranking function, based on random walks, that strongly captures the structural features of the hypergraph-of-entity. We compare this function with two baselines from a traditional Lucene[7] inverted index: TF-IDF and BM25 (with default parameters $k_1 = 1.2$ and $b = 0.75$). Both during indexing and querying, text is preprocessed using an analyzer similar to Lucene's *StandardAnalyzer*, with two main differences: (*i*) stopwords are selected based on the *language-detector* library, using the corresponding dictionaries for the detected language as provided by PostgreSQL 9.6, instead of the default set of English stopwords; (*ii*) tokens with a length inferior to 3 characters are discarded. The ranking function we propose, Random Walk Score, requires the preselection of a set of seed nodes that represent the query. In this section, we describe the seed node selection process, and the Random Walk Score computation approach.

4.2.1 Seed node selection

The seed node selection process can be seen as part of the "organic" process that enables a kind of stochastic semantic tagging of query parts, akin to named entity recognition in queries. Thus, the first step in calculating the Random Walk Score is to map a keyword query to nodes in the hypergraph-of-entity. This process is similar to the graph-of-entity [67], that is, we tokenize the query into unigrams, mapping them to the corresponding term node (if no match exists, the unigram is simply ignored). The term nodes are then expanded to adjacent entity nodes (the seed nodes), which replace them, unless no adjacent entity node exists, resulting in the term node becoming its own seed node. A confidence weight is then calculated for each seed node, measuring the certainty of the node representing the query. See Devezas et al. [67, Section 3.2, Retrieval] for further details.

Ambiguity is not dealt with during seed node selection, but instead during ranking. During seed node selection, we attempt to reach the whole universe of possibilities (i.e., we find the most complete set of candidate entities that might represent the query). During ranking, however, we rely on the overall relations, naturally stored in the hypergraph, for disambiguation. It is not infrequent to do entity linking based on a graph of entities (and sometimes mentions) and their relations [71–73]. What we do here is to use basic substring matching to find a large number of candidates (many times we can have over 1,000 candidate nodes per query). Then, during ranking, while capturing the structure of the hypergraph based on random walks, each candidate will be visited for a given number of times, depending on the link density of the neighborhood of each seed node. Seed nodes act as an open representation of the query. Ambiguity is then solved by cross-referencing all available information through paths in the graph that depart from the seed nodes. Since we also include synonyms in the hypergraph, we aren't even required to consider multiple word senses, as these are naturally solved based on the knowledge of the model. This is why we simply use substring matching. Although such a naive approach to term-entity linking can be improved, we argue that, based on the described strategy, this is only one step towards entity linking. Moreover, based on the cited literature, this is a step that makes sense for our model, where we already capture links between entities and terms

7 https://lucene.apache.org/

(i.e., mentions), which might even be weighted with different degrees of certainty.

4.2.2 Random Walk Score (RWS)

In random walks, steps can be chosen uniformly at random, but we can also establish a bias through weighted hyperedges (which we can also do for graphs) and weighted nodes (used for a random, non-uniform selection of nodes within a hyperedge). We can also vary the length of the walk $\ell \in \{\ell_1, \ell_2, \ldots, \ell_n\}$, as well as the number of repeats (or iterations) $r \in \{r_1, r_2, \ldots, r_m\}$. In particular, we experimented with the configurations given by $\ell \times r \in \{(\ell_1, r_1), (\ell_1, r_2), \ldots, (\ell_n, r_m)\}$. The goal was to measure the impact of increasing the walk length, the number of repeats, or both. The length ℓ constricts or liberates the random walker to wander closer or further apart from the concepts that best represent the query (the seed nodes), while the repeats r improve the certainty of the computed ranking. For evaluation, we used $\ell \in \{2, 3, 4\}$ and $r \in \{10^2, 10^3\}$.

For each seed node, we launched a number r of random walks of length ℓ, storing the number of visits to each hyperedge. Per seed node, we then normalized the number of visits by dividing by the maximum and then multiplying by the seed node confidence weight. These individual scores were then summed for each hyperedge, obtaining a final document hyperedge score. Random walks respected hyperedge direction, as well as node and hyperedge weights, which introduced bias. In practice, this means that we modeled ad hoc document retrieval as a hyperedge ranking problem using biased random walks for ranking. It also means that a similar strategy can be applied to ad hoc entity retrieval by modeling this task as a node ranking problem instead. This demonstrates how the hypergraph-of-entity is a generalizable model that is easily extensible to other entity-oriented search tasks.

5 Evaluation

We experimented with multiple variations of the hypergraph-of-entity, resulting in six different models: (*i*) the base model; (*ii*) the base model extended with *synonym* undirected hyperedges; (*iii*) the base model extended with undirected *context* hyperedges based on word embedding similarities; (*iv*) the base model extended with synonyms and then context; (*v*) the base model extended with context and then synonyms; and (*vi*) the base model extended with synonyms, context and node and hyperedge weights. Table 2 provides an overview of the tested models, showing which nodes and hyperedges were enabled for each model. In particular, it is relevant to notice that the integration order of synonyms and context matters — if we introduce synonyms and only then context, the term vocabulary might increase and word embeddings will also be computed for the synonym terms; on the other hand, if we introduce context and only then synonyms, the opposite might happen, given the word embeddings model has been trained with an external collection whose term vocabulary does not coincide with that from the original collection (this is not the case in the experiments we present here).

In the remainder of this section, we present the INEX 2009 subset we used for evaluation (Section 5.1), we characterize an instance of the hypergraph-of-entity, with all the extensions, including synonyms, context and weights (Section 5.2), we study rank stability, since Random Walk Score is not deterministic (Section 5.3) and, finally, we assess the performance of the hypergraph-of-entity representation and retrieval model, measuring effectiveness, as well as indexing and querying efficiency (Section 5.4).

5.1 INEX 2009 Wikipedia collection

Schenkel et al. [9] have provided an XML collection of Wikipedia articles, annotated with over 5,800 entity classes from the YAGO ontology. The INEX 2009 Wikipedia Collection contains over 2.6 million articles and requires 50.7 GB of disk space, for storage, when uncompressed. The INEX Ad Hoc Track also provided 115 topics from the 2009 occurrence, with 50,725 individual relevance judgments [74], and 107 topics from the 2010 occurrence, with 39,031 individual relevance judgments [40]. Each individual relevance judgment contains the query identifier, the document identifier, the number of relevant characters, the offset of the best entry point (usually the first relevant passage) and offset–length pairs for the relevant passages.

For the INEX 2009 and 2010 Ad Hoc tracks, both topics and relevance judgments were produced by participants. In 2010, only a fraction of the topics (52 out of 107) had associated relevance judgments. Furthermore, only 7 topics were judged by more than one individual, with the remaining topics being judged by a single individual. Despite the reduced number of judges per topic, it is important to notice that individual passages (not documents) were the main object of judgement. This means that every considered document was explored in detail, which was somewhat transposed to the relevance judgments, since

Table 2: Hypergraph-of-entity model overview. Superscript numbers beside the check marks indicate the integration order of a node or hyperedge in the model.

(a) Model nodes. Term nodes can be created based on the document, as well as expanded with synonyms external to the collection and contextually similar terms based on any corpus.

Model	term			entity
	Doc.	Syn.	Cont.	
Base Model	✓	X	X	✓
Syns	✓	✓	X	✓
Context	✓	X	✓	✓
Syns+Context	✓	✓(1)	✓(2)	✓
Context+Syns	✓	✓(2)	✓(1)	✓
Syns+Context+Weights	✓	✓(1)	✓(2)	✓

(b) Model hyperedges. Each hyperedge is depicted with either a tuple of sets (directed) or a single set (undirected). Elements that can be repeated are displayed with a subscript and elements that only appear once have no subscript. We use t to represent term nodes and e to represent entity nodes.

Model	document	contained_in	related_to	synonym	context	weight
	$\{t_n, e_m\}$	$(\{t_n\}, \{e\})$	$\{e_m\}$	$\{t_n\}$	$\{t_n\}$	
Base Model	✓	✓	✓	X	X	X
Syns	✓	✓	✓	✓	X	X
Context	✓	✓	✓	X	✓	X
Syns+Context	✓	✓	✓	✓(1)	✓(2)	X
Context+Syns	✓	✓	✓	✓(2)	✓(1)	X
Syns+Context+Weights	✓	✓	✓	✓(1)	✓(2)	✓

we know exactly which passages led each judge to assign relevance to the document.

Due to the lack of memory for indexing the complete INEX 2009 Wikipedia Collection with the hypergraph-of-entity, which was supported on the serialization of in-memory data structures, we were forced to lower the scale to a smaller subset of the INEX 2009 collection. Accordingly, we prepared a sampling method, based on the topics used for relevance assessment in the INEX 2010 Ad Hoc Track. In order to create the subset, we selected all the 52 topics with relevance judgments, filtering out documents that were not mentioned in the relevance judgments and obtaining a collection of 37,788 documents. While this limits comparison with existing approaches based on the same collection, it still enables us to position the hypergraph-of-entity in regard to Lucene TF-IDF and BM25 baselines.

5.2 Hypergraph characterization

In this section, we characterize the hypergraph-of-entity representation for the INEX 2009 subset, indexed using the base model, with undirected *document* hyperedges, along with *synonym*, *context* and *weight* extensions. We begin by providing overall statistics regarding the number of nodes and hyperedges in the graph. We then analyze the connectivity power of *synonym* and *context* hyperedges, that is, their ability to establish new paths between documents. We finish by providing an overview of the weight distributions for different types of nodes and hyperedges.

Regarding disk space, the base model (the smallest index) required a total of 654 MB for a collection of 203 MB (compressed). Out of the 654 MB, 540 MB were used to store the hypergraph, 100 MB to store node metadata and 15 MB to store hyperedge metadata. On the other hand, the base model extended with synonyms, context and weights (the largest index) required a total of 715 MB of space, out of which 582 MB were used to store the hypergraph, 102 MB to store node metadata, 19 MB to store hyperedge metadata, 8 MB to store node weights and 4.5 MB to store hyperedge weights.

Table 3 shows the number of nodes and hyperedges for each type, also discriminating against direction. As we can see, the total number of hyperedges is significantly lower (almost half) than the number of nodes. This is the op-

Table 3: Number of nodes and hyperedges of the largest index.

(a) Nodes.

Node	Count
term	1,126,685
entity	905,163
Total	2,031,848

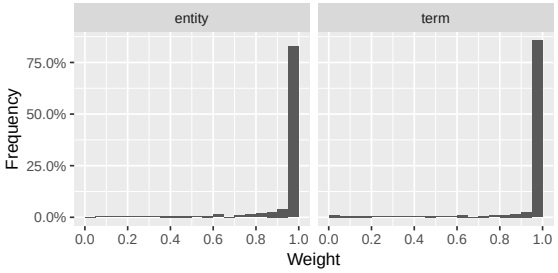
(b) Hyperedges.

Hyperedge	Count
contained_in	784,672
Total directed	784,672
document	37,775
related_to	37,608
synonym	13,749
context	268,505
Total undirected	357,637
Total	1,142,309

posite behavior that we had found in the graph-of-entity, which didn't even include synonyms or contextual information. Most of the nodes in the hypergraph are used to represent terms, closely followed by entities. Most of the hyperedges are directed, specifically used to link terms and entities. Out of the undirected hyperedges, most are used to establish context — we might consider increasing the acceptance threshold for contextually similar terms, when building the word2vec similarity network, in order to lower the number of *context* hyperedges.

Relations of synonymy and contextual similarity were responsible for establishing new connections between documents, which in turn had the potential to improve recall over the base model. We analyzed the base model with synonyms and we found that synonyms established 6,968 new paths between documents, with 219.90 documents linked on average per synonym, with each synonym ranging between 1 and 12,839 linked documents. We did a similar analysis for the base model with context and we found that contextual similarity established 125,333 new paths between documents, with 53.71 documents linked on average through contextual similarity, ranging from 1 to 29,333 linked documents. The significantly higher number of new paths introduced by context, when compared to synonyms, might be explained by the fact that, despite only considering noun synonyms, potentially every word was a candidate for context extraction. On the other hand, we notice that, on average, context established a smaller

(a) Nodes.



(b) Hyperedges.

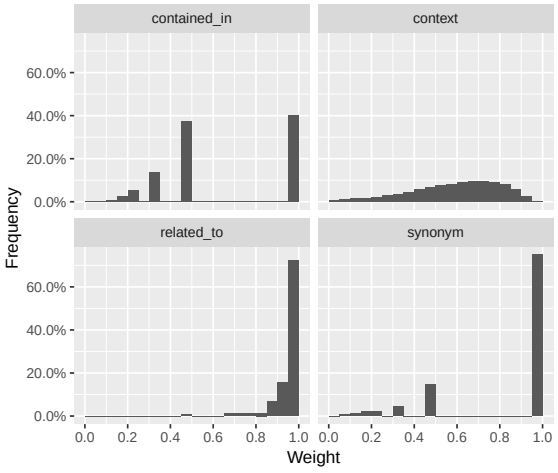


Figure 6: Hypergraph-of-entity weight distributions for INEX 2009.

number of links between documents than synonyms, despite the higher number of paths between each linked document.

Figure 6 illustrates the distribution of node and hyperedge weights. As we can see, the selected weights are generally left skewed, showing a long left tail with most of the values within a range of 0.95 and 1.00 (note that we used a bin width of 0.05). This is less evident for *contained_in* hyperedges and does not happen for *document* hyperedges (not shown in the figure), since their weight is constant (0.5). Both *contained_in* and *synonym* hyperedge weight distributions have multiple missing ranges of values. This means that, granularity-wise these weighting functions are not ideal, regarding their discriminative power. In fact, this is also true for the remaining weighting functions.

5.3 Studying rank stability

While methods based on random walks usually converge to a limiting distribution, there is still a nondeterministic nature to this retrieval approach. This means that the

Table 4: Measuring the stability of Random Walk Score using Kendall's coefficient of concordance (W), for different parameter configurations.

(a) INEX 2009 subset (52 topics; 37,788 documents).

ℓ	r	W
2	10	0.8719
2	50	0.8465
2	100	0.8450
3	10	0.8572
3	50	0.8312
3	100	0.8327
4	10	0.8439
4	50	0.8196
4	100	0.8224

(b) INEX 2009 smaller subset (3 topics; 2,234 documents).

ℓ	r	W	W'
2	100	0.7670	0.8386
2	1000	0.7646	0.9428
2	10000	0.9020	0.9857
3	100	0.7356	0.8733
3	1000	0.7881	0.9617
3	10000	0.9124	0.9901
4	100	0.7144	0.8957
4	1000	0.8178	0.9698
4	10000	0.9203	0.9930

probability distribution of visiting a set of nodes, given a departing set of seed nodes, where random walkers start from, will eventually reach similar values for repeated experiments, given a sufficiently large number of iterations r. Measuring the performance of the Random Walk Score only makes sense when in context with a rank stability analysis, through the measurement of rank convergence, for different runs with the same topic and parameter configuration.

We measured rank stability based on the Kendall's coefficient of concordance (Kendall's W) using fixed configurations of the Random Walk Score as the ranking function. We repeated the same query multiple times, for a given configuration of ℓ and r, and then normalized each ranking list to ensure that they all contained the same set of documents. Missing documents were added to the end of the list, sorted by *doc_id* to ensure consistency in the calculation of Kendall's W, for equivalent rankings, with the same set of missing documents.

Table 4a summarizes the results for $\ell \in \{2, 3, 4\}$ and $r \in \{10, 50, 100\}$, using the geometric mean[8] over 100 repeats for each of the 52 topics of the INEX 2009 subset. For such low values of r, we did not find a significant difference in concordance, beyond a slight indication that a higher walk length ℓ tends to lower the concordance W. This is expected, since the longer the length of the walk, the higher the number of available path choices. More importantly, we found that, even for low values of r, we already achieve a concordance of over 80%. Nonaggregated values for Kendall's W, for each topic and parameter configuration, ranged from 0.7547 to 0.9521, with the first quartile already reaching 0.8030. Standard deviations were under 0.0521, showing stability over different topics. In order to better understand the behavior of concordance for higher values of r, we also replicated the experiment for the smaller subset with $r \in \{100, 1000, 10000\}$. Results, shown in Table 4b, illustrate the overall effect of increasing r — higher values of r result in a higher concordance. Even for low values of r, the results given by the Random Walk Score are already considerably stable, which increases trust that a performance assessment should remain fairly unchanged for different runs with the same parameter configuration. Both tables show the concordance coefficient for $r = 100$, which is lower for the smaller subset. Given the geometric mean was calculated over only three topics, the influence of a single topic was quite impactful. In particular, we found that topic 2010023 (`[ retirement age ]`) resulted in a much lower concordance coefficient, ranging from 0.4544 to 0.7905. Further analysis of the remaining two topics showed that their concordance coefficients were in fact higher than the geometric mean depicts, ranging from 0.8189 to 0.9936. These values were also more in agreement with the experiment for the larger subset, as we can see from the geometric mean W', calculated after removing topic 2010023. Based on the limited but consistent evidence of this analysis, where an incremental behavior of concordance was found for increasing values of r, we chose $r = 10^3$ as a good compromise that should provide an evaluation reliability of approximately 95%.

5.4 Assessing model performance

In the previous section, we have measured the stability of a ranking approach based on random walks. In this section, we will understand how similar parameter configurations affect the performance of the retrieval model. We will then compare the graph-of-entity with the hypergraph-of-entity, regarding effectiveness and efficiency, but also illustrate the difference in number of nodes and edges, particularly regarding the node-edge ratio.

In order to evaluate retrieval over the hypergraph-of-entity, we used the title of each topic from the INEX 2010 Ad Hoc Track as a search query. We then assessed effectiveness based on whether or not retrieved documents contained relevant passages, according to the provided relevance judgments. In order to measure efficiency, we also collected indexing and search times, as to understand the cost of using such a hypergraph-based representation, as

8 We used the geometric mean, since it is less sensitive to outliers and always smaller than the arithmetic mean, thus providing a more conservative result. However, for this particular case, the difference between arithmetic and geometric means was negligible.

well as different parameter configurations for the Random Walk Score.

We tested each of the variations presented in Table 2, assessing the effectiveness of the Random Walks Score, using a combination of parameter configurations based on low walk lengths and high walk repeats, according to the intuition that closer nodes to the seeds (and therefore to the query) lead to more relevant documents/entities and that a higher number of repeats leads to convergence and therefore trustworthy results. We obtained the best hypergraph-of-entity MAP for the base model extended with synonyms, contextually similar terms and weights, with $\ell = 2$ and $r = 10^3$ (cf. Table 5a) — we verified that increasing values of r suggested an increasing and plateauing performance. None of the hypergraph-of-entity variations were able to surpass the Lucene baselines, reaching MAP values between 0.0811 and 0.0884, when compared to 0.1689 for TF-IDF and 0.3269 for BM25. The best hypergraph-of-entity model according to GMAP, MAP and precision was "Syns+Context+Weights", however the "Base Model" without extensions was able to reach the best results for NDCG@10 and P@10. We carried a Student's t-test for the 28 pairs of models, comparing average precisions for MAP and individual P@10 values per topic, using a p-value of 0.05. Results showed that the difference in MAP, as well as P@10, was statistically significant for TF-IDF and BM25, as well as for any Lucene baseline and any hypergraph-of-entity model, but not among different versions of our model.

The introduction of weights shows the flexibility of the model, in the sense that it is able to easily support the boosting of terms and entities, as well as the boosting of documents and other relations, in order to assign, for instance, a degree of certainty to each piece of information. Experiments also showed that the higher the walk length ℓ, the worse the retrieval effectiveness. This is, by design, the expected behavior, since the further apart nodes are from the seed nodes (which represent the query), the less related to the query they are and thus the less relevant they are. The best recall for the hypergraph-of-entity was obtained for "Context+Syns" (0.8148), which was close to the baselines (0.8476 for TF-IDF and 0.8598 for BM25). The Geometric Mean Average Precision (GMAP) was included in Table 5a because it is less affected by outliers than MAP, thus providing additional insight. Through the comparison of GMAP and MAP, it becomes evident that a small number of topics are driving MAP up for the hypergraph-of-entity, despite many individual topics resulting in a low average precision — in some cases achieving values as low as zero (e.g., for topic 2010006 on the best "Context" model).

In Table 5b, we find the indexing and search times for the runs with the best MAP per variation. The hypergraph-of-entity took 2.9 times longer to index than Lucene, when comparing "Syns" with "Lucene", as well as between 18.8 times longer to query (best case scenario, for the "Syns" model with $\ell = 2$ and $r = 10^2$ and Lucene TF-IDF) and 1127 times longer to query (worst case scenario for "Syns+Context+Weights" with $\ell = 4$ and $r = 10^3$ and Lucene BM25 with $k_1 = 1.2$ and $b = 0.75$). Given the notable difference in efficiency between the weighed and non-weighted versions, it might be a good compromise to use the "Base Model" with $\ell = 2$ and $r = 10^3$, which is the most effective model when considering the top 10. Overall, search time was shown to range roughly between 9 and 23 minutes for $l = 4$ and $r = 10^3$ runs, with MAP scores between 0.06 and 0.08 and a coefficient of concordance around 0.82. However, if we consider $l = 2$ and a lower value $r = 10^2$, search time will drop to a range roughly between 22 seconds and 1 minute, with MAP scores of roughly 0.06 and a coefficient of concordance dropping to around 0.77. This means that we can achieve comparable effectiveness, while significantly increasing efficiency, despite compromising the concordance of multiple similar runs with the same parameter configuration (i.e., the ranking function won't converge).

5.4.1 Comparing graph-of-entity and hypergraph-of-entity

In order to better understand the differences in performance between the graph-of-entity and the hypergraph-of-entity, we were required to further reduce the size of the test collection. In particular, we used a smaller subset of the INEX 2009 Wikipedia Collection, so that we were able to generate the graph-of-entity in a timely manner. Sampling was based on a selection of 10 topics uniformly at random, filtering out documents that were not mentioned in the relevance judgments and obtaining a collection of 7,487 documents (80% smaller than the subset based on 52 topics).

Table 6 compares the effectiveness and efficiency of graph-of-entity and hypergraph-of-entity, using Lucene as a baseline. For the graph-of-entity, we used the Entity Weight (EW) as the ranking function. For the hypergraph-of-entity, we used the base model (i.e., without synonyms, context or weights) and the Random Walk Score (RWS) as the ranking function. As we can see in Table 6a, hypergraph-of-entity is overall more effective than graph-of-entity, except when considering the macro averaged precision (Prec.). As shown in Table 6b, hypergraph-of-entity

Table 5: Best overall parameter configuration according to the mean average precision.

(a) Effectiveness (highest values for Lucene and Hypergraph-of-Entity in bold; differences in MAP are not statistically significant, except between the Lucene baselines and the hypergraph-of-entity indexes).

Index	Ranking	GMAP	MAP	Prec.	Rec.	NDCG@10	P@10
Lucene	TF-IDF	0.1345	0.1689	**0.0650**	0.8476	0.2291	0.2346
	BM25	**0.2740**	**0.3269**	0.0647	**0.8598**	**0.5607**	**0.5250**
Hypergraph-of-Entity: Random Walk Score ($\ell = 2$, $r = 10^3$)							
Base Model	RWS	0.0285	0.0864	0.0219	0.8003	**0.1413**	**0.1269**
Syns	RWS	0.0281	0.0840	0.0225	0.8099	0.1301	0.1231
Context	RWS	0.0134	0.0811	0.0220	0.8027	0.1218	0.1192
Syns+Context	RWS	0.0299	0.0837	0.0236	0.8069	0.1310	0.1231
Context+Syns	RWS	0.0296	0.0814	0.0242	**0.8148**	0.1256	0.1250
Syns+Context+Weights	RWS	**0.0313**	**0.0884**	**0.0274**	0.8059	0.1256	0.1154

(b) Efficiency (lowest times for Lucene and Hypergraph-of-Entity in bold).

Index	Ranking	Indexing Time		Search Time	
		Avg./Doc	Total	Avg./Query	Total
Lucene	TF-IDF	2.16ms	1m 21s 382ms	**1s 148ms**	**59s 698ms**
	BM25			1s 220ms	1m 03s 461ms
Hypergraph-of-Entity: Random Walk Score ($\ell = 2$, $r = 10^3$)					
Base Model	RWS	6.52ms	4m 05s 612ms	**3m 22s 826ms**	**2h 55m 47s**
Syns	RWS	**6.22ms**	**3m 54s 587ms**	3m 31s 038ms	3h 02m 54s
Context	RWS	6.35ms	3m 59s 446ms	3m 35s 623ms	3h 06m 52s
Syns+Context	RWS	6.29ms	3m 57s 264ms	3m 33s 000ms	3h 04m 36s
Context+Syns	RWS	6.33ms	3m 58s 659ms	3m 36s 487ms	3h 07m 37s
Syns+Context+Weights	RWS	6.52ms	4m 05s 984ms	10m 55s 590ms	9h 28m 11s

is also considerably more efficient than graph-of-entity, taking only 53s 992ms to index when compared to 1h 38m for graph-of-entity. When analyzing search time for the hypergraph-of-entity, we can see that there is a trade-off between effectiveness and efficiency that can be controlled through parameter r. For higher values, like $r = 10^4$, we reach a MAP score of 0.1689 but search time is a lot higher than the graph-of-entity (13m 4s when compared to 21s 557ms). On the other hand, for lower values, like $r = 10^1$ or even $r = 10^2$, where we reach MAP scores of 0.0485 and 0.1118 respectively, search time is lower than the graph-of-entity (943ms and 11s 134ms when compared to 21s 557ms). Additionally, by lowering the value of r, we also lower the rank stability, but even for $r \in \{10, 50, 100\}$ we were able to achieve coefficients of concordance of around 0.85 (cf. Table 4a), which might be an acceptable compromise efficiency-wise. The gain in indexing speed is particularly influenced by the growth in number of (hyper)edges when compared to the number of nodes. While the graph-of-entity has 10 times more edges than nodes, the hypergraph-of-entity has 2.4 times less edges than nodes. We also carried a Student's t-test for the 21 pairs of models, comparing average precisions for MAP and individual P@10 values per topic, using a p-value of 0.05. Results showed that the difference in MAP was statistically significant for TF-IDF and BM25, for BM25 and any hypergraph-of-entity model (except for $r = 10$), and for graph-of-entity and the Lucene baselines. When considering P@10, behavior was similar, except for TF-IDF and any hypergraph-of-entity model, where the difference in P@10 was not statistically significant.

6 Conclusion

We have proposed a unified representation model for the representation and retrieval of text and knowledge, assess-

Table 6: Graph-of-entity (GoE) vs hypergraph-of-entity (HGoE) with $\ell = 2$.

(a) Effectiveness (highest values for Lucene and graph-based models in bold).

Index	Ranking	GMAP	MAP	Prec.	Rec.	NDCG@10	P@10
Lucene	TF-IDF	0.1540	0.1710	0.1389	0.8007	0.2671	0.2800
	BM25	**0.2802**	**0.2963**	**0.1396**	**0.8241**	**0.5549**	**0.5000**
GoE	EW	0.0003	0.0399	**0.1771**	0.2233	0.1480	0.1500
HGoE	RWS($r = 10^1$)	0.0000	0.0485	0.0734	0.3085	0.1229	0.1200
	RWS($r = 10^2$)	0.0546	0.1118	0.0342	0.7554	0.1474	0.1500
	RWS($r = 10^3$)	0.1017	0.1492	0.0199	0.9122	**0.2074**	**0.2200**
	RWS($r = 10^4$)	**0.1224**	**0.1689**	0.0167	**0.9922**	0.1699	0.1700

(b) Efficiency (lowest times for Lucene and graph-based models in bold).

Index	Ranking	Indexing Time (Total)	Search Time (Avg./Query)	Nodes	Edges
Lucene	TF-IDF	**27s 769ms**	**209ms**	N/A	N/A
	BM25		316ms		
GoE	EW	1h 38m	21s 557ms	981,647	9,942,647
HGoE	RWS($r = 10^1$)	**53s 922ms**	**943ms**	607,213	253,154
	RWS($r = 10^2$)		11s 134ms		
	RWS($r = 10^3$)		1m 17s 540ms		
	RWS($r = 10^4$)		13m 04s 057ms		

ing it in regard to the task of ad hoc document retrieval. We have provided a comprehensive survey of entity-oriented and semantic retrieval tasks, proposing a joint representation for text and knowledge, capable of supporting multiple retrieval tasks, without the need to change the model. We have used the hypergraph data structure as an alternative solution for capturing higher-order dependencies in documents, entities and their relations. We presented and assessed the performance of a base model, as well as multiple combinable extensions, using synonyms provided by WordNet, context provided by word2vec word embeddings similarity, and node and hyperedge weighting functions. We proposed the Random Walk Score as a method for relevance scoring and as a retrieval model that closely depends on the structure of the hypergraph, thus providing the flexibility to change and improve the representation model without the need to repeatedly revise the ranking function. Finally, we evaluated several aspects of the model, characterizing the obtained hypergraph, studying rank stability and identifying the parameter configurations that best ensure the concordance of repeated queries with the same configuration. In some cases, we obtained MAP scores comparable to Lucene TF-IDF, while capturing and integrating heterogeneous information in a generalized representation model that provides explicit semantics and extreme flexibility in the definition of n-ary relations, such as synonyms and context, as well as subsumption or hierarchical relations. We also showed that the hypergraph-of-entity is significantly more efficient than the graph-of-entity in indexing time and that it can also be configured, through parameter r, for faster search times with only a small penalty in effectiveness. One of the more evident limitations of the model is the lack of consideration for document length. Although verbosity is mitigated (term repetitions are not considered), vocabulary diversity in long documents that cover multiple topics is still a problem (a kind of pivoted document length normalization is required). Despite its performance limitations, particularly when compared to state-of-the-art approaches, hypergraph-based representations have the potential to more naturally model our cognition process, unlocking increasingly intelligent information retrieval systems as we study and approach the brain.

6.1 Future work

There are several pending improvements over the hypergraph-of-entity, even regarding basic tasks, such as plural removal, stemming or lemmatization. At this stage, however, we used an inclusive policy to avoid discarding potentially relevant information. Therefore, in the future, we will focus on ways to decrease the size of the model. For instance, we would like to measure the impact of pruning nodes and hyperedges from the hypergraph, independently for each type of node and hyperedge, as well as based on different weight thresholds. Another alternative would be to prune similar hyperedges (e.g., based on the Jaccard index). The goal is to understand how much we can improve efficiency until performance starts to decrease. In the same line, we would also like to explore alternative approaches to generating the context similarity network, using different word embedding strategies, as well as avoiding non-optimal algorithms, like *k*-nearest neighbors, to obtain the top similar terms. An alternative would easily be the usage of pivots for an approximated measurement of similarity [75].

Apart from reducing the model by pruning redundancies, we would, on the other hand, like to extend it with synonyms for verbs, adjectives and adverbs, measuring the impact in effectiveness, and understanding whether the usage of *synsets* for nouns had been sufficient. Another interesting idea, that has been proven to improve query understanding [76], is the usage of dependency parsing. It would be interesting to extract term dependencies from the documents in a collection, building a dependency graph and integrating these relations into the hypergraph (like we did for the word2vec similarity network). The idea is that it might indirectly improve query understanding, even for simple keyword queries, and thus positively impact the overall retrieval effectiveness.

While we have focused on improving the efficiency of graph-of-entity by defining a new hypergraph-of-entity model, there are still scalability issues to be tackled. In particular, we would like to asses how the model scales over datasets like the complete INEX 2009 Wikipedia Collection or even the DBLP co-authorship network. We predict that, as the size of the collection increases, efficiency problems will become more prominent and we think this can be mitigated with different approaches to the computation of random walks, for instance based on fingerprinting, as described by Fogaras et al. [77] or Chakrabarti [27].

While we have proposed the usefulness of a hypergraph-based model to capture subsumption and hierarchical relations, we haven't properly assessed the impact that such decision has in retrieval effectiveness. This is something that we will focus on in the future. Additionally, in order to assess the generality of the model, we intend to also implement and measure the effectiveness of other tasks from entity-oriented search, including ad hoc entity retrieval, entity list completion and related entity finding, over a common representation model. Finally, regarding node and hyperedge weighting functions, there are still many open questions that we aim to answer. In particular, it is not clear what the best approach to weighting is, whether weights can be learned automatically and whether such weighting models should be dependent on the target domain or query intent. In the future, we will also explore these questions, in particular in conjunction with the tasks of entity list completion and related entity finding, which always provide a target type for querying.

Acknowledgements: José Devezas is supported by research grant PD/BD/128160/2016, provided by the Portuguese national funding agency for science, research and technology, Fundação para a Ciência e a Tecnologia (FCT), within the scope of Operational Program Human Capital (POCH), supported by the European Social Fund and by national funds from MCTES.

We would like to thank our colleagues at FEUP InfoLab, João Rocha for suggesting a switch to memory-based representations and Joana Rodrigues for patiently listening to rambles about hypergraphs.

References

[1] Gomes F., Devezas J., Figueira Á., Temporal visualization of a multidimensional network of news clips, In: Advances in Information Systems and Technologies, Springer, 2013, 157–166

[2] Belkin N. J., Croft W. B., Information filtering and information retrieval: Two sides of the same coin?, In: Communications of the ACM, 1992, 35(12), 29–38

[3] Bautin M., Skiena S., Concordance-based entity-oriented search, In: IEEE/WIC/ACM Conference on Web Intelligence (WI'07), 2007, 2–5

[4] Blanco R., Lioma C., Graph-based term weighting for information retrieval, In: Information Retrieval, 2012, 15(1), 54–92

[5] Rousseau F., Vazirgiannis M., Graph-of-word and TW-IDF: New approach to ad hoc IR, In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, ACM, 2013, 59–68

[6] Bu J., Tan S., Chen C., Wang C., Wu H., Zhang L., He X., Music recommendation by unified hypergraph: Combining social media information and music content, In: Proceedings of the 18th International Conference on Multimedia, Firenze, Italy, October 25-29, 2010, 391–400

[7] Xiong C., Callan J., Liu T., Word-entity duet representations for document ranking, In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017, 763–772
[8] Bast H., Buchhold B., Haussmann E., Semantic search on text and knowledge bases, In: Foundations and Trends® in Information Retrieval, 2016, 10(2-3), 119–271
[9] Schenkel R., Suchanek F. M., Kasneci G., YAWN: A semantically annotated Wikipedia XML corpus, In: Datenbanksysteme in Business, Technologie und Web (BTW 2007), 12. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), Proceedings, 7.-9. März 2007, Aachen, Germany, 2007, 277–291
[10] Luhn H. P., A statistical approach to mechanized encoding and searching of literary information, In: IBM Journal of Research and Development, 1957, 1(4), 309–317
[11] Sparck Jones K., A statistical interpretation of term specificity and its application in retrieval, In: Journal of Documentation, 1972, 28(1), 11–21
[12] Robertson S. E., Walker S., Jones S., Hancock-Beaulieu M., Gatford M., Okapi at TREC-3, In: Proceedings of The Third Text Retrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994, 109–126
[13] Ponte J. M., Croft W. B., A language modeling approach to information retrieval, In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, August 24-28 1998, 275–281
[14] Amati G., van Rijsbergen C. J., Probabilistic models of information retrieval based on measuring the divergence from randomness, In: ACM Transactions on Information Systems, 2002, 20(4), 357–389
[15] Kraaij W., Westerveld T., Hiemstra D., The importance of prior probabilities for entry page search, In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, August 11-15, 2002, 27–34
[16] Westerveld T., KraaijW., Hiemstra D., Retrieving web pages using content, links, URLs and anchors, In: Proceedings of The Tenth Text Retrieval Conference, TREC 2001, Gaithersburg, Maryland, USA, November 13-16, 2001
[17] Brin S., Page L., The anatomy of a large-scale hypertextual web search engine, In: Computer Networks, 1998, 30(1-7), 107–117
[18] Badache I., Boughanem M., A priori relevance based on quality and diversity of social signals, In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015, 731–734
[19] Badache I., Boughanem M., Emotional social signals for search ranking, In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017, 1053–1056
[20] Macdonald C., Ounis I., Voting for candidates: Adapting data fusion techniques for an expert search task, In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006, 387–396
[21] Fang Y., Si L., Related entity finding by unified probabilistic models, In: World Wide Web, 2015, 18(3), 521–543
[22] Davison B. D., Topical locality in the web, In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, July 24-28, 2000, 272–279
[23] Raiber F., Kurland O., Exploring the cluster hypothesis, and cluster-based retrieval, over the web, In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, HI, USA, October 29 - November 02, 2012, 2507–2510
[24] Hogan A., Harth A., Decker S., ReConRank: A scalable ranking method for semantic web data with context, In: Proceedings of Second International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006), in conjunction with International Semantic Web Conference (ISWC 2006), 2006
[25] Balmin A., Hristidis V., Papakonstantinou Y., ObjectRank: Authority-based keyword search in databases, In: (e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3, 2004, 564–575
[26] Nie Z., Zhang Y., Wen J., Ma W., Object-level ranking: Bringing order to web objects, In: Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, May 10-14, 2005, 567–574
[27] Chakrabarti S., Dynamic personalized PageRank in entity-relation graphs, In: Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, May 8-12, 2007, 571–580
[28] Delbru R., Toupikov N., Catasta M., Tummarello G., Decker S., Hierarchical link analysis for ranking web data, In: The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 – June 3, 2010, Proceedings, Part II, 2010, 225–239
[29] Raviv H., Kurland O., Carmel D., Document retrieval using entity-based language models, In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pisa, Italy, July 17-21, 2016, 65–74
[30] Neumayer R., Balog K., Nřrvĺg K., On the modeling of entities for ad-hoc entity search in the web of data, In: Advances in Information Retrieval - 34th European Conference on IR Research, ECIR 2012, Barcelona, Spain, April 1-5, 2012, Proceedings, 2012, 133–145
[31] Lin B., Rosa K. D., Shah R., Agarwal N., LADS: Rapid development of a learning-to-rank based related entity finding system using open advancement, In: The First International Workshop on Entity-Oriented Search (EOS), 2011
[32] Schuhmacher M., Dietz L., Ponzetto S. P., Ranking entities for web queries through text and knowledge, In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, Melbourne, VIC, Australia, October 19-23, 2015, 1461–1470
[33] Chen J., Xiong C., Callan J., An empirical study of learning to rank for entity search, In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, July 17-21, 2016, 737–740
[34] Tonon A., Demartini G., Cudré-Mauroux P., Combining inverted indices and structured search for ad-hoc object retrieval, In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, OR, USA, August 12-16, 2012, 125–134
[35] Cao L., Guo J., Cheng X., Bipartite graph based entity ranking for related entity finding, In: Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence, Campus Scientifique de la Doua, Lyon, France, August 22-27, 2011, 2011,

130–137
[36] Raviv H., Kurland O., Carmel D., The cluster hypothesis for entity oriented search, in: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2013, 841–844
[37] Bron M., Balog K., de Rijke M., Example based entity search in the web of data, In: Advances in Information Retrieval – 35th European Conference on Information Retrieval, ECIR 2013, Moscow, Russia, March 24-27, 2013, Proceedings, 2013, 392–403
[38] Pound J., Mika P., Zaragoza H., Ad-hoc object retrieval in the web of data, In: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, 771–780
[39] Devezas J., Coelho F., Nunes S., Ribeiro C., Music Discovery: Exploiting TF-IDF to boost results in the long tail of the tags distribution, 2013
[40] Arvola P., Geva S., Kamps J., Schenkel R., Trotman A., Vainio J., Overview of the INEX 2010 ad hoc track, In: Comparative Evaluation of Focused Retrieval - 9th International Workshop of the Inititative for the Evaluation of XML Retrieval, INEX 2010, Vugh, The Netherlands, December 13-15, 2010, Revised Selected Papers, 2010, 1–32
[41] Demartini G., Iofciu T., de Vries A. P., Overview of the INEX 2009 entity ranking track, In: Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2009, Brisbane, Australia, December 7-9, 2009, Revised and Selected Papers, 2009, 254–264
[42] Clarke C. L. A., Craswell N., Soboroff I., Overview of the TREC 2009 web track, In: Proceedings of The Eighteenth Text Retrieval Conference, TREC 2009, Gaithersburg,Maryland, USA, November 17-20, 2009
[43] Balog K., Serdyukov P., de Vries A. P., Overview of the TREC 2011 entity track, In: Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011
[44] Campinas S., Ceccarelli D., Perry T. E., Delbru R., Balog K., Tummarello G., The Sindice-2011 dataset for entity-oriented search in the web of data, In: The First International Workshop on Entity-Oriented Search (EOS), 2011, 26–32
[45] Dkaki T., Mothe J., Truong Q. D., Passage retrieval using graph vertices comparison, In: Proceedings of the 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, Shanghai, China, December 16-18, 2007, 71–76
[46] Page L., Brin S., Motwani R., Winograd T., The PageRank citation ranking: Bringing order to the web, Technical report, Stanford InfoLab, 1999
[47] Khurana U., Deshpande A., Eflcient snapshot retrieval over historical graph data, In: Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013), Brisbane, Australia, April 8-12, 2013, 997–1008
[48] Martins B., Silva M. J., A Graph-Ranking Algorithm for Geo-Referencing Documents, In: Proceedings of the Fifth IEEE International Conference on Data Mining, Houston, Texas, USA, 27-30 November, 2005, 741–744
[49] Zhu Y., Yan E., Song I., A natural language interface to a graph-based bibliographic information retrieval system, In: Data & Knowledge Engineering, 2017, 111, 73–89
[50] Blanco R., Mika P., Vigna S., Effective and efficient entity search in RDF data, In: The Semantic Web – ISWC2011 – 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I, 2011, 83–97
[51] Bendersky M., Croft W. B., Modeling higher-order term dependencies in information retrieval using query hypergraphs, In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, Oregon, USA, 2012, 941–950
[52] Xiong S., Ji D., Query-focused multi-document summarization using hypergraph-based ranking, In: Information Processing & Management, 2016, 52(4), 670–681
[53] Haentjens Dekker R., Birnbaum D. J., It's more than just overlap: Text As Graph, In: Proceedings of Balisage: The Markup Conference 2017, 19, 2017
[54] Cattuto C., Schmitz C., Baldassarri A., Servedio V. D. P., Loreto V., Hotho A., Grahl M., Stumme G., Network properties of folksonomies, In: AI Communications, 2007, 20(4), 245–262
[55] Seidman S. B., Structures induced by collections of subsets: A hypergraph approach, In: Mathematical Social Sciences, 1981, 1(4), 381–396
[56] Tan S., Bu J., Chen C., Xu B., Wang C., He X., Using rich social media information for music recommendation via hypergraph model, In: ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) – Special Section on ACM Multimedia 2010 Best Paper Candidates, and Issue on Social Media, 2011, 7S(1), 22
[57] McFee B., Lanckriet G. R. G., Hypergraph models of playlist dialects, In: Proceedings of the 13th International Society for Music Information Retrieval Conference, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012, 343–348
[58] Theodoridis A., Kotropoulos C., Panagakis Y., Music recommendation using hypergraphs and group sparsity, In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, May 26-31, 2013, 56–60
[59] von Neumann J., The computer and the brain, Yale University Press, 2012
[60] Sporns O., Networks of the brain, MIT press, 2010
[61] Davison E. N., Schlesinger K. J., Bassett D. S., Lynall M.-E., Miller M. B., Grafton S. T., Carlson J. M., Brain network adaptability across task states, In: PLOS Computational Biology, 2015, 11(1), 1–14
[62] Jie B., Wee C.-Y., Shen D., Zhang D., Hyper-connectivity of functional networks for brain disease diagnosis, In: Medical Image Analysis, 2016, 32, 84–100
[63] Gu S., Yang M., Medaglia J. D., Gur R. C., Gur R. E., Satterthwaite T. D., Bassett D. S., Functional hypergraph uncovers novel covariant structures over neurodevelopment, In: Human Brain Mapping, 2017, 38(8), 3823–3835
[64] Zhang B. T., Random hypergraph models of learning and memory in biomolecular networks: Shorter-term adaptability vs. longer term persistency, In: 2007 IEEE Symposium on Foundations of Computational Intelligence, 2007, 344–349
[65] Goertzel B., Patterns, hypergraphs and embodied general intelligence, In: The 2006 IEEE International Joint Conference on Neural Network Proceedings, 2006, 451–458
[66] Bellaachia A., Al-Dhelaan M., Random walks in hypergraph, In: Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods, Venice Italy, 2013, 187–194
[67] Devezas J., Lopes C. T., Nunes S., FEUP at TREC 2017 OpenSearch track: Graph-based models for entity-oriented search, In: The Twenty-Sixth Text REtrieval Conference Proceedings (TREC 2017),

Gaithersburg, MD, USA, 2017
[68] Devezas J., Nunes S., Graph-based entity-oriented search: imitating the human process of seeking and cross referencing information, In: ERCIM News, Special Issue: Digital Humanities, 2017, 111, 13–14
[69] Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., Distributed representations of words and phrases and their compositionality, In: Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, United States, 2013, 2, 3111–3119
[70] Robertson S., Understanding inverse document frequency: On theoretical arguments for IDF, In: Journal of Documentation, 2004, 60(5), 503–520
[71] Alhelbawy A., Gaizauskas R., Graph ranking for collective named entity disambiguation, In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, 2, 75–80
[72] Hoffart J., Yosef M. A., Bordino I., Fürstenau H., Pinkal M., Spaniol M., Taneva B., Thater S., Weikum G., Robust disambiguation of named entities in text, In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, 782–792
[73] Moro A., Raganato A., Navigli R., Entity linking meets word sense disambiguation: A unified approach, In: Transactions of the Association for Computational Linguistics, 2014, 2, 231–244
[74] Geva S., Kamps J., Lehtonen M., Schenkel R., Thom J. A., Trotman A., Overview of the INEX 2009 ad hoc track, In: Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2009, Brisbane, Australia, December 7-9, 2009, Revised and Selected Papers, 2009, 4–25
[75] Coelho F., Ribeiro C., Automatic illustration with cross-media retrieval in large-scale collections, In: 2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI), 2011, 25–30
[76] Liu J., Pasupat P.,Wang Y., Cyphers S., Glass J., Query understanding enhanced by hierarchical parsing structures, In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, 2013, 72–77
[77] Fogaras D., Rácz B., Csalogány K., Sarlós T., Towards scaling fully personalized PageRank: Algorithms, lower bounds, and experiments, In: Internet Mathematics, 2011, 2(3), 333–358