RESEARCH

Characterizing the hypergraph-of-entity and the structural impact of its extensions

José Devezas* and Sérgio Nunes

*Correspondence: jld@fe.up.pt
INESC TEC and Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, s/n, 4200-465 Porto, PT
Full list of author information is available at the end of the article

Abstract

The hypergraph-of-entity is a joint representation model for terms, entities and their relations, used as an indexing approach in entity-oriented search. In this work, we characterize the structure of the hypergraph, from a microscopic and macroscopic scale, as well as over time with an increasing number of documents. We use a random walk based approach to estimate shortest distances and node sampling to estimate clustering coefficients. We also propose the calculation of a general mixed hypergraph density measure based on the corresponding bipartite mixed graph. We analyze these statistics for the hypergraph-of-entity, finding that hyperedge-based node degrees are distributed as a power law, while node-based node degrees and hyperedge cardinalities are log-normally distributed. We also find that most statistics tend to converge after an initial period of accentuated growth in the number of documents. We then repeat the analysis over three extensions — materialized through *synonym*, *context*, and *tf_bin* hyperedges — in order to assess their structural impact in the hypergraph. Finally, we focus on the application-specific aspects of the hypergraph-of-entity, in the domain of information retrieval. We analyze the correlation between the retrieval effectiveness and the structural features of the representation model, proposing ranking and anomaly indicators, as useful guides for modifying or extending the hypergraph-of-entity.

Keywords: Hypergraph-of-entity; Hypergraph analysis; Information retrieval; Indexing; Combined data; Representation model; Characterization

1 Introduction

Complex networks have frequently been studied as graphs, but only recently has attention been given to the study of complex networks as hypergraphs [1]. The hypergraph-of-entity [2] is a hypergraph-based model used to represent combined data [3, §2.1.3]. That is, it is a joint representation of corpora and knowledge bases, integrating terms, entities and their relations. It attempts to solve, by design, the issues of representing combined data through inverted indexes and quad indexes. The hypergraph-of-entity, together with its random walk score [2, §4.2.2], is also an attempt to generalize several tasks of entity-oriented search. This includes ad hoc document retrieval and ad hoc entity retrieval, as well as the recommendation-alike tasks of related entity finding and entity list completion. However, there is a tradeoff. On one side, the random walk score acts as a general ranking function. On the other side, it performs below traditional baselines like TF-IDF (term frequency $\times$ inverted document frequency). Since ranking is particularly dependent on the structure of

the hypergraph, a characterization is a fundamental step towards improving the representation model and, with it, the retrieval performance.

Accordingly, our focus was on studying the structural features of the hypergraph. This is a task that presents some challenges, both from a practical sense and from a theoretical perspective. While there are many tools [4, 5] and formats [6, 7] for the analysis and transfer of graphs, hypergraphs still lack clear frameworks to perform these functions, making their analysis less trivial. Even formats like GraphML [7] only support undirected hypergraphs. Furthermore, there is still an ongoing study of several aspects of hypergraphs, some of which are trivial in graph theory. For example, the adjacency matrix is a well-established representation of a graph, however recent work is still focusing on defining an adjacency tensor for representing general hypergraphs [8]. As a scientific community, we have been analyzing graphs since 1735 and, even now, innovative ideas in graph theory are still being researched [9]. However, the concept of hypergraph is much younger, dating from 1970 [10], and thus there are still many open challenges and contribution opportunities.

In this work, which is an extended version of our previous characterization work [11], we take a practical application of hypergraphs in the domain of information retrieval, the hypergraph-of-entity, as an opportunity to establish a basic framework for the analysis of hypergraphs. We expand on our previous work by analyzing the impact of two extensions (synonymy, and contextual similarity), that had previously been defined for this representation model [2], and we also introduce and characterize a new extension, based on the idea of segmenting the document into different sets of terms according to discretizations of term frequency (TF-bins, or term frequency bins). The main contributions of this work are the following:

- Analysis of multiple versions of real-world hypergraph data structures being developed for information retrieval;
- Proposal of a practical analysis framework for hypergraphs;
- Proposal of estimation approaches for the computation of shortest paths and clustering coefficients in hypergraphs;
- Proposal of a computation approach for the density of general mixed hypergraphs based on a corresponding bipartite graph representation;
- Example of an application in the context of information retrieval, where structural features were measured over different hypergraph-based models and presented in context with the performance of each model.

The remainder of this document is organized as follows. In Section 2, we begin by providing an overview on the analysis of the inverted index, knowledge bases and hypergraphs, covering the three main aspects of the hypergraph-of-entity. In Section 3.2, we describe our characterization approach, covering shortest distance estimation with random walks and clustering coefficient estimation with node sampling, as well as proposing a general mixed hypergraph density formula by establishing a parallel with the corresponding bipartite mixed graph. In Section 5, we present the results of a characterization experiment of the hypergraph-of-entity for a subset of the INEX (INitiative for the Evaluation of XML Retrieval) 2009 Wikipedia collection and, in Section 6, we explore the effect of including synonyms, contextual similarity, or TF-bins in the structure of the hypergraph. In Section 7, we assess the retrieval effectiveness of the representation model, analyzing the correlations between the evaluation metrics and the structural features (Section 7.1), and

proposing ranking and anomaly indicators based on our conclusions (Section 7.2). Finally, in Section 8, we close with the conclusions and future work.

2 Reference work

The hypergraph-of-entity is a representation model for indexing combined data, jointly modeling unstructured textual data from corpora and structured interconnected data from knowledge bases. As such, before analyzing a hypergraph from this model, we surveyed existing literature on inverted index analysis, as well as knowledge base analysis. We then surveyed literature specifically on the analysis of hypergraphs, particularly focusing on statistics like clustering coefficient, shortest path lengths and density.

2.1 Analyzing inverted indexes

There are several models based on the inverted index that combine documents and entities [12, 13] and that are comparable with the hypergraph-of-entity. There has also been work that analyzed the inverted index, particularly regarding query evaluation speed and space requirements [14, 15].

Voorhees [14] compared the efficiency of the inverted index with the top-down cluster search. She analyzed the storage requirements of four test collections, measuring the total number of documents and terms, as well as the average number of terms per document. She then analyzed the disk usage per collection, measuring the number of bytes for document vectors and the inverted index. Finally, she measured CPU time in number of instructions and the I/O time in number of data pages accessed at least once, also including the query time in seconds.

Zobel et al. [15] took a similar approach to compare the inverted index and signature files. First, they characterized two test collections, measuring size in megabytes, number of records and distinct words, as well as the record length, and the number of words, distinct words and distinct words without common terms per record. They also analyzed disk space, memory requirements, ease of index construction, ease of update, scalability and extensibility.

For the hypergraph-of-entity characterization, we do not focus on measuring efficiency, but rather on studying the structure and size of the hypergraph.

2.2 Analyzing knowledge bases

Studies have been made to characterize the entities and triples in knowledge bases. In particular, given the graph structure of RDF (resource description framework), we are interested in understanding which statistics are relevant for instance to discriminate between the typed nodes.

Halpin [16] took advantage of Microsoft's *Live.com* query log to reissue entity and concept queries over their FALCON-S semantic web search engine. They then studied the results, characterizing their source, triple structure, RDF and OWL (web ontology language) classes and properties, and the power-law distributions of the number of URIs, both returned as results and as part of the triples linking to the results. They focused mostly on measuring the frequency of different elements or aggregations (e.g., top-10 domain names for the URIs, most common data types, top vocabulary URIs).

Ge et al. [17] defined an object link graph based on the graph induced by the RDF graph, based on paths linking objects (or entities), as long as they are either direct or established through blank nodes. They then studied this graph for the Falcons Crawl 2008 and 2009 datasets (FC08 and FC09), which included URLs from domains like bio2rdf.org or dbpedia.org. They characterized the object link graph based on density, using the average degree as an indicator, as well as connectivity, analyzing the largest connected component and the diameter. They repeated the study for characterizing the structural evolution of the object link graph, as well its domain-specific structures (according to URL domains). Comparing two snapshots of the same data enabled them to find evidence of the scale-free nature of the network. While the graph almost doubled in size from FC08 to FC09, the average degree remained the same and the diameter actually decreased.

Fernandez et al. [18] focused on studying the structural features of RDF data, identifying redundancy through common structural patterns, proposing several specific metrics for RDF graphs. In particular, they proposed several subject and object degrees, accounting for the number of links from/to a given subject/object (outdegree and indegree), the number of links from a $\langle subject, predicate \rangle$ (partial outdegree) and to a $\langle predicate, object \rangle$ (partial indegree), the number of distinct predicates from a subject (labeled outdegree) and to an object (labeled indegree), and the number of objects linked from a subject through a single predicate (direct outdegree), as well as the number of subjects linking to an object through a single predicate (direct indegree). They also measured predicate degree, outdegree and indegree. They proposed common ratios to account for shared structural roles of subjects, predicates and objects (e.g., subject-object ratio). Global metrics were also defined for measuring the maximum and average outdegree of subject and object nodes for the whole graph. Another analysis approach was focused on the predicate lists per subject, measuring the ratio of repeated lists and their degree, as well as the number of lists per predicate. Finally, they also defined several statistics to measure typed subjects and classes, based on the *rdf:type* predicate.

While we study a hypergraph that jointly represents terms, entities and their relations, we focus on a similar characterization approach, that is more based on structure and less based on measuring performance.

2.3 Analyzing hypergraphs

Hypergraphs [10] have been around since 1970. While this concept was introduced by Claude Berge on this year, there had been other contributions surrounding the topic, namely in extremal graph and set theory. Post-1970, the work by Erdös [19] and Brown et al. [20] illustrates the intersection between extremal graph theory and hypergraph theory, while, pre-1970, we can also find contributions like Sperner's theorem [21], in extremal set theory, or the Turán number [22, 23], in extremal graph theory. Interestingly, hypergraphs have remained somewhat fringe in network science, perhaps due to Paul Erdös resistance to the concept [10]:

> *At the Balatonfüred Conference (1969), P. Erdös and A. Hajnal asked us why we would use hypergraphs for problems that can be also formulated in terms of graphs. The answer is that by using hypergraphs, one deals with generalizations of familiar concepts. Thus, hypergraphs can be used to simplify as well as to generalize.*

Although Erdös himself, who was interested in exploring the representation of graphs using set intersections [24], also studied hypergraph problems, he avoided this designation, only sparsely using it [20]:

> *By an r-graph we mean a fixed set of vertices together with a class of unordered subsets of this fixed set, each subset containing exactly r elements and called an r-tuple. In the language of Berge [10] this is a simple uniform hypergraph of rank r.*

Hypergraphs are data structures that can capture higher-order relations. As such, they either present conceptually different or multiple counterparts to the equivalent graph statistics. Take for instance the node degree. While graphs only have a node degree, indegree and outdegree, hypergraphs can also have a hyperedge degree, which is the number of nodes in a hyperedge [25]. The hyperedge degree also exists for directed hyperedges, in the form of a tail degree and a head degree[1]. The tail degree is based on the cardinality of the source node set and the head degree is based on the cardinality of the target node set. In this work we will rely on the degree, clustering coefficient, average path length, diameter and density to characterize the hypergraph-of-entity.

Building on the work by Gallo et al. [26], who extended Dijkstra's algorithm to hypergraphs, and the work by Ausiello et al. [27], who tackled the same problem using a dynamic approach, Gao et al. [28] have also proposed two algorithms for computing shortest paths in hypergraphs. The first, HyperEdge-based Dynamic Shortest Path (HE-DSP), like Gallo et al., proposed an extension to Dijkstra's algorithm. The second, Dimension Reduction Dynamic Shortest Path (DR-DSP), relied on an induced graph with the same vertex set, adding weighted edges when a hyperedge containing the two vertices exists in the corresponding hypergraph, while selecting the minimum weight over all available hyperedges for the pair of vertices.

In this work, we focus on approximated computation approaches, which are useful for large-scale hypergraphs. Ribeiro et al. [29] proposed the use of multiple random walks to find shortest paths in power law networks. They found that random walks had the ability to observe a large fraction of the network and that two random walks, starting from different nodes, would intersect with a high probability. Glabowski et al. [30] contributed with a shortest path computation solution based on ant colony optimization, clearly structuring it as pseudocode, while providing several configuration options. Parameters included the number of ants, the influence of pheromones and other data in determining the next step, the speed of evaporation of the pheromones, the initial, minimum and maximum pheromone levels, the initial vertex and an optional end vertex. Li [31] studied the computation of shortest paths in electric networks based on random walk models and ant colony optimization, proposing a current reinforced random walk model inspired by the previous two. In this work, we also use a random walk based approach to approximate shortest paths and estimate the average path length and diameter of the graph.

Gallagher and Goldberg [32, Eq.4] provide a comprehensive review on clustering coefficients for hypergraphs. The proposed approach for computing the clustering

[1] Tail and head is used in analogy to an arrow, not a list.

coefficient in hypergraphs accounted for a pair of nodes, instead of a single node, which is more frequent in graphs. Based on these two-node clustering coefficients, the node cluster coefficient was then calculated. Two-node clustering coefficients measured the fraction of common hyperedges between two nodes, through the intersection of the incident hyperedge sets for the two nodes. It then provided different kinds of normalization approaches, either based on the union, the maximum or minimum cardinality, or the square root of the product of the cardinalities of the hyperedge sets. The clustering coefficient for a node was then computed based on the average two-node clustering coefficient for the node and its neighbors.

The codegree Turán density [33] $\gamma(\mathcal{F})$ can be computed for a family $\mathcal{F}$ of k-uniform hypergraphs, also known as k-graphs. It is calculated based on the codegree Turán number co-ex$(n, \mathcal{F})$ — the extremal number based on the codegree in a hypergraph, instead of the degree in a graph — which takes as parameters the number of nodes n and the family $\mathcal{F}$ of k-graphs. In turn, the codegree Turán number is calculated based on the minimum number of nodes, taken from all sets of $r-1$ vertices of each hypergraph H_n that, when united with an additional vertex, will form a hyperedge from H. The codegree density for a family $\mathcal{F}$ of hypergraphs is then computed based on $\limsup_{n\to\infty} \frac{\text{co-ex}(n,\mathcal{F})}{n}$. Since this was the only concept of density we found associated with hypergraphs or, more specifically, a family of k-uniform hypergraphs, we opted to propose our own density formulation (Section 3.2). Furthermore, the hypergraph-of-entity is a single general mixed hypergraph. In other words, it is not a family of hypergraphs, it contains hyperedges of multiple degrees (it's not k-uniform, but general) and it contains undirected and directed hyperedges (it's mixed). Accordingly, we propose a density calculation based on the counterpart bipartite graph of the hypergraph, where hyperedges are translated to bridge nodes.

3 Methodology

In this section, we introduce general concepts and definitions, formally providing mathematical support for this analysis. Next, we present the characterization methodology and propose approaches to estimate shortest distances, clustering coefficients and density. Finally, we describe the methodology for a practical application of this analysis framework in the domain of information retrieval.

3.1 General concepts and definitions

We provide a mathematical framework, where we formalize several concepts and definitions, including relevant classes of hypergraphs, as well as useful properties and statistics, that we rely upon across this manuscript.

3.1.1 Classes of hypergraphs

In this section we formally define hypergraph, distinguishing between undirected, directed and mixed, as well as uniform and general.

Definition 3.1 (Hypergraph) Let v be a vertex and V be a set of vertices such that $v \in V$, with $n = |V|$ being the number of vertices. Let $E = E_U \cup E_D$ be the set of all hyperedges, where E_U represents the subset of undirected hyperedges $e_U \in E_U$ and E_D the subset of directed hyperedges $e_D \in E_D$, with $m = |E_U| + |E_D| = |E|$

being the total number of hyperedges. Let also a set $e_U \subseteq V$ be an undirected hyperedge and a tuple of sets $e_D = (t, h)$ be a directed hyperedge formed by a tail set $t \subseteq V$ (source) and a head set $h \subseteq V$ (target). A hypergraph is then a tuple $H = (V, E)$.

Definition 3.2 (Hypergraph direction) Under this notation, a hypergraph $H = (V, E)$ is said to be:

- Undirected, when $E = E_U$ or, equivalently, $E_D = \emptyset$;
- Directed, when $E = E_D$ or, equivalently, $E_U = \emptyset$;
- Mixed, when $E_U \neq \emptyset \wedge E_D \neq \emptyset$.

Definition 3.3 (Hypergraph uniformity) A uniform or k-uniform hypergraph is characterized by all of its hyperedges being defined over the same number k of vertices. For an undirected hyperedge e_U it means $|e_U| = k$, while for a directed hyperedge $e_D = (t, h)$ it means $|t| + |h| = k$.

On the other hand, a non-uniform hypergraph is said to be a general hypergraph, which contains hyperedges of diverse cardinalities.

* Please refer to Banerjee and Char [34] for more information on directed uniform hypergraphs.

Definition 3.4 (Hyperedge incidence) Let $v \in V$ have the following sets of incident hyperedges:

- $E_v = E_{U_v} \cup E_{D_v}$ as the set of all incident hyperedges to v, ignoring direction;
- $E_v^- = E_{U_v} \cup E_{D_v}^-$ as the set of all incoming hyperedges to v;
- $E_v^+ = E_{U_v} \cup E_{D_v}^+$ as the set of all outgoing hyperedges from v.

3.1.2 Hypergraph statistics

In this section, we formally describe the hypergraph statistics that we rely upon for our analysis framework. In particular we describe the different degrees that can be computed for a vertex, the cardinalities of hyperedges, the diameter and average shortest path length, the clustering coefficient, and the density.

Definition 3.5 (Vertex-based vertex degree) Let $d_v(v)$ be the degree of a vertex measured based on the number of adjacent vertices.

Vertex-based degree (ignoring direction) is given by:

$$d_v(v) = \sum_{e_U \in E_{U_v}} |e_U| + \sum_{(t,h) \in E_{D_v}} (|t| + |h|)$$

Vertex-based indegree is given by:

$$d_v^-(v) = \sum_{e_U \in E_{U_v}} |e_U| + \sum_{(t,h) \in E_{D_v}^-} |t|$$

And vertex-based outdegree is given by:

$$d_v^+(v) = \sum_{e_U \in E_{U_v}} |e_U| + \sum_{(t,h) \in E_{D_v}^+} |h|$$

Definition 3.6 (Hyperedge-based vertex degree) Let $d_h(v)$ be the degree of a vertex measured based on the number of incident hyperedges.

Hyperedge-based degree (ignoring direction) is given by:

$$d_h(v) = |E_v|$$

Hyperedge-based indegree is given by:

$$d_h^-(v) = |E_v^-|$$

And hyperedge-based outdegree is given by:

$$d_h^+(v) = |E_v^+|$$

Definition 3.7 (Hyperedge cardinality) Let $c(e)$ be the cardinality of a hyperedge measured based on the number of nodes it contains. Let e_U be an undirected hyperedge and $e_D = (t, h)$ be a directed hyperedge.

Undirected hyperedge cardinality is given by:

$$c(e_U) = |e_U|$$

Directed hyperedge cardinality is given by:

$$c(e_D) = |t| + |h|$$

In order to index hyperedges based on their number of nodes, we also use the notation E_U^a to represent sets of undirected hyperedges of cardinality $a = |e_U|$, as well as $E_D^{a,b}$ to represent sets of directed hyperedges with a tail of size $a = |t|$ and a head of size $b = |h|$.

Definition 3.8 (Diameter / avg. short. path len.) Let L be the set of shortest path lengths between all pairs of connected nodes. Let $\ell_{u,v} \in L$ be the length of the shortest path between nodes u and v from the vertex set V. For $e_{U_i}, e_{U_j} \in E_U$ and $e_{D_i}, e_{D_j} \in E_D$, we define L as follows:

$$L = \left\{ \ell_{u,v} : \ u \in e_{U_i} \wedge v \in e_{U_j} \ \ \vee \ \ u \in t \wedge (t, \cdot) \in e_{D_i} \wedge v \in h \wedge (\cdot, h) \in e_{D_j} \right\}$$

The diameter is then given by:

$$\max L$$

And the average shortest path length is given by:

$$\frac{1}{|L|} \sum_{\ell_{i,j} \in L} \ell_{i,j}.$$

Definition 3.9 (Clustering coefficient) The clustering coefficient measures the degree to which nodes tend to agglomerate in dense groups. We compute this metric based on the following approach by Gallagher and Goldberg [32]. Let $E_v = E_{U_v} \cup E_{D_v}$ be the set of incident hyperedges to v, ignoring direction. Let $N(v)$ be the set of all vertices adjacent to v (i.e., sharing a hyperedge, while ignoring direction).

The clustering coefficient $cc(u, v)$ for a pair of nodes u and v is given by:

$$cc(u, v) = \frac{|E_u \cap E_v|}{|E_u \cup E_v|}$$

The clustering coefficient $cc(v)$ for a single node v is given by:

$$cc(v) = \frac{1}{|N(v)|} \sum_{u \in N(v)} cc(u, v)$$

And the clustering coefficient $cc(H)$ for the hypergraph is given by:

$$cc(H) = \frac{1}{|V|} \sum_{v \in V} cc(v)$$

Definition 3.10 (Density) We transform a hypergraph $H = (V, E)$ into its corresponding bipartite graph $G_H = (\mathcal{V}, \mathcal{E})$, using the density of G_H as an indicator of density for H.

The vertices $\mathcal{V}$ of G_H are based on the vertices V and hyperedges E from H and are given by:

$$\mathcal{V} = V \cup \{v_e : e \in E\}$$

The edges $\mathcal{E} = \mathcal{E}_U \cup \mathcal{E}_D$ of G_H are established based on all pairs of vertices connected by a hyperedge $E = E_U \cup E_D$ from H.

The undirected edges $\mathcal{E}_U$ of G_H are given by:

$$\mathcal{E}_U = \{(u, v_e), (v_e, w) : e \in E_U \wedge u \in e \wedge w \in e\}$$

And the directed edges $\mathcal{E}_D$ of G_H are given by:

$$\mathcal{E}_D = \{(u, v_e), (v_e, w) : e = (t, h) \in E_D \wedge u \in t \wedge w \in h\}$$

Density $D(H)$, or simply D, is then given by:

$$D = D(G_H) = \frac{2|\mathcal{E}_U| + |\mathcal{E}_D|}{2|\mathcal{V}|\,(|\mathcal{V}| - 1)}$$

3.2 Hypergraph characterization approach

Graphs can be characterized at a microscopic, mesoscopic and macroscopic scale. The microscopic analysis is concerned with statistics at the node-level, such as the degree or clustering coefficient. The mesoscopic analysis is concerned with statistics and patterns at the subgraph-level, such as communities, network motifs or graphlets. The macroscopic analysis is concerned with statistics at the graph-level, such as average clustering coefficient or diameter. In this work, our analysis of the hypergraph is focused on the microscopic and macroscopic scales. We compute several statistics for the whole hypergraph, as well as for snapshot hypergraphs that depict growth over time. Some of these statistics are new to hypergraphs, when compared to traditional graphs. For instance, nodes in directed graphs have an indegree and an outdegree. However, nodes in directed hypergraphs have four degrees, based on incoming and outgoing nodes, as well as on incoming and outgoing hyperedges. While in graphs all edges are binary, leading to only one other node, in hypergraphs hyperedges are n-ary, leading to multiple nodes, and thus different degree statistics. While some authors use 'degree' to refer to node and hyperedge degrees [35, §4][25, §Network Statistics in Hypergraphs], in this work we opted to use the 'degree' designation when referring to nodes and the 'cardinality' designation when referring to hyperedges. This is to avoid any confusion for instance between an "hyperedge-induced" node degree and a hyperedge cardinality.

We analyze the base model, as well as three models based on the synonyms, contextual similarity and TF-bins extensions. For the full hypergraph of each of the four models, we compute the following global statistics:

- Number of nodes, in total and per type;
- Number of hyperedges, in total, per direction, and per type;
- Average degree;
- Average clustering coefficient;
- Average path length;
- Diameter;
- Density.

We also plot the following distributions for the full hypergraph:

- Node degree distributions per node type:
  - Node-based node degree;
  - Hyperedge-based node degree.
- Hyperedge cardinality distributions per hyperedge type.

Then, we define a temporal analysis framework based on an increasing number of documents (i.e., time passes as documents are added to the hypergraph-of-entity index). We prepare several snapshots, with a different number of documents each, for each of the four models. We then compute and plot the following statistics for each snapshot, showing its evolution as the number of documents increases:

- Average node degree over time;
- Average hyperedge cardinality over time;
- Average diameter and average path length over time;
- Average clustering coefficient over time;
- Average density over time.
- Size over time:
  - Number of nodes;
  - Number of hyperedges;
  - Space in disk;
  - Space in memory.

Finally, we also measure the run time for several operations, in order to understand the efficiency cost and the evolution of its behavior for an increasing number of documents:

- Index creation time;
- Global statistics computation time;
- Node degrees computation time;
- Hyperedge cardinalities computation time.

In order to support large-scale hypergraphs, we compute the average path length, diameter, clustering coefficient, and density using approximated strategies. We estimate shortest distances based on random walks, the clustering coefficient based on node sampling, and the density based on a bipartite graph induced from the hypergraph, although without the need to explicitly create this graph. The following sections will detail these approaches.

3.2.1 Estimating shortest distances with random walks

Ribeiro et al. [29] found that, in power law networks, there is a high probability that two random walk paths, usually starting from different nodes, will intersect and share a small fraction of nodes. We took advantage of this conclusion, adapting it to a hypergraph, in order to compute a sample of shortest paths and their length, used to estimate the average path length and diameter. We considered two (ordered) sets $S_1 \subset V$ and $S_2 \subset V$ of nodes sampled uniformly at random, each of size $s = |S_1| = |S_2|$. We then launched r random walks of length ℓ from each pair of nodes S_1^i and S_2^i. For a given pair of random walk paths, we iterated over the nodes in the path starting from S_1^i, until we found a node in common with the path starting from S_2^i. At that point, we merged the two paths based on the common node, discarding the suffix of the first path and the prefix of the second path. We computed the length of these paths, keeping only the minimum length over the r repeats. As the number of iterations r increased, we progressively approximated the shortest path for the pair of nodes. Despite the inherent estimation error, this method can be used to study even large-scale hypergraphs — precision can be controlled by tuning the number of sampled nodes and random walks, which will eventually lead to convergence for large values. This approach enabled us to generate a sample of approximated shortest path lengths, which could be used to compute the estimated diameter (its maximum) and the estimated average path length (its mean), in a scenario where high precision is not critical. This is true for instance for a quick or initial analysis of a hypergraph. Given the repeated research iterations over the hypergraph-of-entity and the multitude of tests carried over this model, a quick estimation approach is ideal.

3.2.2 Estimating clustering coefficients with node sampling

In a graph, the clustering coefficient is usually computed for a single node and averaged over the whole graph. As shown by Gallagher and Goldberg [32, §I.A.], in hypergraphs the clustering coefficient is computed, at the most atomic level, for a pair of nodes. The clustering coefficient for a node is then computed based on the averaged two-node clustering coefficients between the node and each of its neighbors (cf. Gallagher and Goldberg [32, Eq.4]). Three options were provided for calculating the two-node clustering coefficient, one of them based on the Jaccard index between the neighboring hyperedges of each node [32, Eq.1], which we use in this work. While a global understanding of the clustering coefficient is useful for characterizing overall local connectivity in the hypergraph, the existence of a random hypergraph generation model, like the Watts–Strogatz model [36] for graphs, would provide further interpretations at a mesoscale. We leave this open and instead focus on the macroscale.

Continuing with the philosophy of large-scale hypergraph support in our analysis framework, as opposed to computing the clustering coefficient for all nodes, we estimated the clustering coefficients for a smaller sample $S \subseteq V$ of nodes. Furthermore, for each sampled node $s_i \in S$, we also sampled its neighbors $N_S(s_i)$ for computing the two-node clustering coefficients. We then applied the described equations to obtain the clustering coefficients for each node s_i and a global clustering coefficient based on the overall average. For $S = V \wedge N_S(s_i) = N(s_i)$, being N_S the sampled

neighbors and N the full set of neighbors, we would obtain the exact clustering coefficient. Again, this approach offers two parameters that can be controlled as a tradeoff between between efficiency and effectiveness.

3.2.3 Computing the density of general mixed hypergraphs

A general mixed hypergraph is general (or non-uniform) in the sense that its hyperedges can contain an arbitrary number of vertices, and it is mixed in the sense that it can contain hyperedges that are either undirected and directed. We compute a hypergraph's density by analogy with its corresponding bipartite graph, which contains all nodes from the hypergraph, along with connector nodes representing the hyperedges.

Consider the hypergraph $H = (V, E)$, with $n = |V|$ nodes and $m = |E|$ hyperedges. Also consider the set of all undirected hyperedges E_U and directed hyperedges E_D, where $E = E_U \cup E_D$. Their subsets E_U^k and $E_D^{k_1,k_2}$ should also be respectively considered, where E_U^k is the subset of undirected hyperedges with k nodes and $E_D^{k_1,k_2}$ is the subset of directed hyperedges with k_1 tail (source) nodes, k_2 head (target) nodes and $k = k_1 + k_2$ nodes, assuming the hypergraph only contains directed hyperedges between disjoint tail and head sets. This means that the union of $E_U = E_U^1 \cup E_U^2 \cup E_U^3 \cup \ldots$ and $E_D = E_D^{1,1} \cup E_D^{1,2} \cup E_D^{2,1} \cup E_D^{2,2} \cup \ldots$ forms the set of all hyperedges E. We use it as a way to distinguish between hyperedges with different degrees. This is important because, depending on the degree k, the hyperedge will contribute differently to the density, when considering the corresponding bipartite graph. For instance, one undirected hyperedge with degree $k = 4$ will contribute with four edges to the density. Accordingly, we derive the density of a general mixed hypergraph as shown in Equation 1.

$$D = \frac{2\sum_k k|E_U^k| + \sum_{k_1,k_2}(k_1 + k_2)|E_D^{k_1,k_2}|}{2(n+m)(n+m-1)} \tag{1}$$

In practice, this is nothing more than a comprehensive combination of the density formulas for undirected and directed graphs. On one side, we consider the density of a mixed graph that should result of the combination of an undirected simple graph and a directed simple graph. That is, each pair of nodes can be connected, at most, by an undirected edge and two directed edges of opposing directions. On the other side, we use hypergraph notation to directly obtain the required statistics from the corresponding mixed bipartite graph, thus calculating the analogous density for a hypergraph.

3.3 Contextualizing through a practical application

In order to study the usefulness of the analysis framework that we propose, we explore it in the context of an information retrieval application. In particular, our use case is based on ad hoc document retrieval (leveraging entities). For this retrieval task, given a keyword query, the goal is to retrieve and rank the documents that best answer the information need of the user. As an entity-oriented search task, the approach must take into account entities, mentioned in documents, and their relations to improve retrieval performance. Evaluation is then done based on a

set of topics (whose title is usually used as the keyword query), along with a set of relevance judgments, containing relevance grades assigned by the judges on multiple retrieved documents.

In this experiment, we attempt to identify individual properties of the hypergraph that correlate with the retrieval performance scores that we compute. We identify indicator properties that help us rank our models by effectiveness, as well as identify models that might be low performers. Although this is also a contribution of this work, we consider it to be secondary, compared to the analysis framework that we propose.

4 Data modeling

In this section, we begin by presenting the test collection that we use to build several hypergraphs based on the hypergraph-of-entity model. Then, we provide an overview of the hypergraph-of-entity, describing the construction approach of the hypergraphs that we study, and a description of the random walk score. Finally, we present the motivation to characterize this unified model for entity-oriented search.

4.1 INEX 2009 Wikipedia collection

In this work, we characterize hypergraphs built based on different versions of the hypergraph-of-entity model, relying upon the INEX 2009 Wikipedia collection [37]. We also explore an application in the domain of information retrieval, where assessment is dependent on the topics and relevance judgments from the INEX 2010 Ad Hoc track. In this section, we describe this test collection, including the main dataset and the subset prepared for the analysis and information retrieval application, as well as the associated topics and relevance judgments, also known as qrels (query relevance set).

Main dataset The INEX 2009 Wikipedia collection[2] is an XML version of articles from the English Wikipedia, based on the dump from October 8, 2008, and incorporating semantic annotations from the 2008-w40-2 version of YAGO (Yet Another Great Ontology)[3]. Like DBpedia[4], YAGO is a semantic knowledge base, containing structured data from Wikipedia, WordNet and GeoNames. The INEX 2009 Wikipedia collection is provided in multiple `tar.bz2` archives that contain nearly 2.7 million articles, requiring 50.7 GB of disk space when uncompressed and only 5.5 GB when compressed, and it relies on over 5,800 classes from YAGO, including people, movies, and cites. Each XML document also contains links to other articles, corresponding to the hyperlinks found in the Wikipedia dump. In total, there are nearly 102 million XML elements in the collection. In order to build the hypergraph, we rely on the text nodes of the `<bdy>` element, as well as on the `<link>` elements to create semantic triples that capture the different entity names based on mentions. The structure of the hypergraph will be further detailed in Section 4.2. For our application to information retrieval (Section 7), we also rely on the qrels for the INEX 2010 Ad Hoc track[5], in a study to determine possible correlations between

[2] `https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/software/inex/`

[3] `https://yago-knowledge.org/`

[4] `https://wiki.dbpedia.org/`

[5] `https://inex.mmci.uni-saarland.de/data/documentcollection.html`

Table 1: Hypergraph-of-entity nodes and hyperedges for the base model and the extensions.

Type	Description	Observation
	Nodes	
term	Represents a single word from the original document.	In this work, the preprocessing pipeline includes: sentence segmentation; lower case filtering; replacement of URL, time, money and number expressions with a common placeholder, each; stemming via porter stemmer.
entity	Represents an entity from the list of extracted entities and/or provided triples.	For the INEX collection, each mention to an entity is modeled through this type of node (we consider disambiguation to be a part of the ranking).
	Hyperedges (Base Model)	
document	Represents a document through the set of all its terms and entities.	Undirected hyperedge.
related_to	Represents a semantic relation between multiple entities.	Undirected hyperedge. In this implementation, the relation is derived from all triples in the collection, by grouping by subject.
contained_in	Represents a relation between a set of terms and an entity.	Directed hyperedge. In this implementation, this relation exists between terms that are a part of an entity name or mention and the corresponding entity node.
	Hyperedges (Extensions)	
synonym	Represents a relation of synonymy between a set of terms.	Undirected hyperedge. Present in the Synonyms model. The first synset from WordNet 3.0 is obtained for each noun term, missing terms are added to the model and the hyperedge is created.
context	Represents a relation of contextual similarity between a set of terms.	Undirected hyperedge. Present in the Contextual similarity model. This is computed based on the top similar terms according to word2vec embeddings.
tf_bin	Represents a sets of terms within the same term frequency interval, for a given document.	Undirected hyperedge. Present in the TF-bins model. The number of TF-bins per document is a parameter that can be set during indexing.

the effectiveness of ad hoc document retrieval (leveraging entities) and the properties of the hypergraphs. Provided relevance grades are binary (0 for irrelevant and 1 for relevant).

INEX 2009 10T-NL subset Due to the space and time complexity of the hypergraph-of-entity, we prepared a smaller subset of the INEX 2009 Wikipedia collection, that we could use to circumvent performance issues. In fact, characterizing the corresponding hypergraph-of-entity for a smaller subset will enable us to identify weaknesses in our model that could help us improve the scalability or retrieval effectiveness of future versions. The subset was created based on a random sample of 10 topics ('10T'). In particular, the following topics were considered: 2010003, 2010014, 2010023, 2010032, 2010038, 2010040, 2010049, 2010057, 2010079, 2010096. We then included only documents mentioned in the relevance judgments for the selected topics, optionally considering linked documents (in this case, we did not include linked documents — accordingly, 'NL' stands for "no linked").

4.2 Hypergraph-of-entity representation and retrieval model

The hypergraph-of-entity [2] is a unified model for entity-oriented search. It provides a joint representation for corpora and knowledge bases, through a general mixed hypergraph, containing the types of nodes and hyperedges described in Table 1. Ranked retrieval then relies on a universal ranking function, called the random

walk score, that supports multiple entity-oriented search tasks, by simply controlling the input (e.g., keyword or entity query) and output (e.g., documents or entities): ad hoc document retrieval (leveraging entities), ad hoc entity retrieval, and entity list completion.

4.2.1 Representation model

In this work, we explore multiple hypergraph-of-entity versions of the representation model, including:

- Base model, with *term* and *entity* nodes, and *document*, *related_to* and *contained_in* hyperedges;
- Synonyms model, extending the base model with *synonym* hyperedges;
- Contextual similarity model, extending the base model with *context* hyperedges;
- TF-bins models, extending the base model with *tf_bin* hyperedges, according to the selected number of bins (we experiment with 2 to 10 TF-bins).

Each of the analyzed hypergraphs is built by indexing the INEX 2009 Wikipedia collection, based on the text in the `<bdy>` element and semantic triples formed from `<link>` elements, where the subject is the entity described by the current article and the object is the entity described by the linked article. No predicates are considered, as these are not a part of the model.

Synonyms are context-based. Our goal is for disambiguation of context to happen naturally through the additional information provided by terms and entities grouped through *document* hyperedges, as well as from the *related_to* hyperedges between entities. A given synonym will be more frequently visited by a random walk, when a higher number of paths from the query nodes (which establish context) also lead the walker there.

Contextual similarity is defined for terms that are frequently surrounded by similar sequences of terms, i.e., that are used in a similar context. In order to establish a relation of contextual similarity, we rely on word2vec [38] to obtain a distributed representation of words (i.e., a word embedding — a vector of latent features that semantically represents a word). After obtaining the word embeddings, we simply use a k-nearest neighbors approach to find the k most similar words based on cosine similarity, ensuring a similarity above 0.5. The original term, as well as the k-nearest neighbors are then grouped in a *context* hyperedge.

Term frequency bins (or TF-bins) are computed as follows. For each document, we calculate the term frequency and, for a given number of bins n, we compute the percentiles $P_n = \{100\frac{x}{n} \mid x \in \mathbb{Z}^+ \wedge x \leq n\}$, assigning them the weight $w(x) = \frac{x}{n}$. So, for example, if we consider $n = 4$ bins, then we compute the percentiles $P_4 = \{25, 50, 75, 100\}$, resulting in four values of TF (term frequency). Let us for instance consider the following term frequency for 10 documents: $1, 1, 1, 1, 2, 2, 2, 2, 2, 3$. This would result in the value 1 for the 25 percentile, 2 for the 50 and 75 percentiles, and 3 for the 100 percentile. We would then form the TF intervals $]0, 1]$, $]1, 2]$, $]2, 2]$ and $]2, 3]$, with the interval $]2, 2]$ having no matches in $\mathbb{Z}^+$, thus making it redundant. Per document, and for each non-empty interval, a weighted hyperedge was then created to group terms with a similar term frequency (i.e., within the same TF-bin). This can be used by the ranking function, to issue biased random walks,

controlling the flow in a way that the walker will be driven towards documents with a higher TF for the query terms.

4.2.2 Retrieval model

Ranked retrieval is done based on RWS (random walk score). A query can be formed by any combination of the elements represented in the hypergraph, as can the results that we score. Most commonly, we define the following three tasks:

- Ad hoc document retrieval, which takes a keyword query as input (mapped to a set of term nodes) and ranks a set of documents, through their hyperedges, as output;
- Ad hoc entity retrieval, which also takes a keyword query as input, but instead ranks a set of entities, through their nodes, as output;
- Entity list completion, which takes an entity query as input (mapped to a set of entity nodes) and ranks a set of entities, through their nodes, as output.

In this work, however, we only explore the task of ad hoc document retrieval, to illustrate an practical application of our hypergraph analysis framework. Regardless of the retrieval task, the random walk score always runs over the whole hypergraph, scoring each node and hyperedge, based on multiple random walks launched from a set of seed nodes that are either a direct or an expanded representation of the query. The random walk score $RWS(\ell, r, \Delta_{nf}, \Delta_{ef}, exp.)$ is a universal ranking function where, for each seed node, r random walks of length ℓ are launched. Each node and hyperedge has a zero score by default, storing the number of visits by random walkers. This is then normalized between zero and one, by dividing by the overall maximum number of visits. The probability resulting from the normalization is then multiplied by the probability of the seed node being a good representative of the query — this is given by the fraction of query nodes linked to the seed node (always one for a direct representation of the query) and the total number of neighbors of the seed node [2, §4.2]. The parameters Δ_{nf} and Δ_{ef} are not used in the experiments we present here and thus are set to zero. The *exp.* parameter determines whether we use a direct or an expanded query representation — we set it to *false*, thus disabling expansion and using the existing nodes for the terms in the query as a the seed nodes.

4.3 Why characterize the hypergraph-of-entity?

While the hypergraph-of-entity is able to serve as a unified framework for entity-oriented search, it is still severely outperformed by baselines like Lucene TF-IDF and BM25 (cf. Table 6). As such, we rely on hypergraph analysis to gain further insights on the structure, and to identify possible changes that could lead to a more effective and efficient model. Briefly, the reasons to characterize the hypergraph-of-entity are the following:

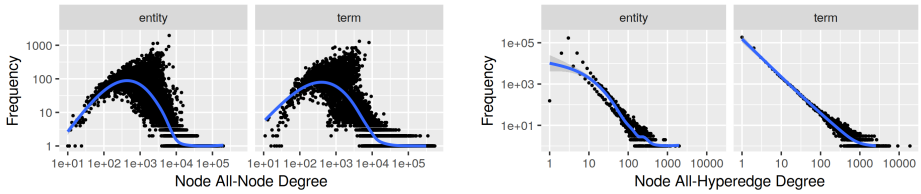
- It supports decision making in the design iterations over the retrieval model;
- Statistics like the average path length will help us tune the random walk score length parameter, and the clustering coefficient will help us understand how many repeated random walks to issue;
- Understanding the evolution of the hypergraph, as the number of documents increases, also gives us insights on how to measure the impact of the pruning

Table 2: Global statistics for the base model.

Statistic	Value
Nodes	607,213
term	323,672
entity	283,541

Statistic	Value
Hyperedges	253,154
Undirected	14,938
document	7,484
related_to	7,454
Directed	238,216
contained_in	238,216

Statistic	Value
Avg. Degree	0.8338
Avg. Clustering Coefficient	0.1148
Avg. Path Length	8.3667
Diameter	17
Density	3.88e-06



(a) Based on connected nodes. (b) Based on connected hyperedges.

Figure 1: Node degree distributions for the base model (log-log scale).

that we apply to the model (e.g., removing redundancies, or retaining only document keywords).

5 Analyzing the hypergraph-of-entity base model

We indexed a subset of the INEX 2009 Wikipedia collection [37] given by the 7,487 documents appearing in the relevance judgments of 10 random topics. We then computed global statistics (macroscale), local statistics (microscale) and temporal statistics. Temporal statistics were based on an increasingly larger number of documents, by creating several snapshots of the index, through a 'limit' parameter, until all documents were considered.

Global statistics In Table 2, we present several global statistics about the hypergraph-of-entity, in particular the number of nodes and hyperedges, discriminated by type, the average degree, the average clustering coefficient, the average path length, the diameter and the density. The average clustering coefficient was computed based on a sample of 5,000 nodes and a sample of 100,000 neighbors for each of those nodes. The average path length and the diameter were computed based on a sample of shortest distances between 30 random pairs of nodes and the intersections of 1,000 random walks of length 1,000 launched from each element of the pair. Finally, the density was computed based on Equation 1. As we can see, for the 7,487 documents the hypergraph contains 607,213 nodes and 253,154 hyperedges of different types, an average degree lower than one (0.83) and a low clustering coefficient (0.11). It is also extremely sparse, with a density of 3.9e−06. Its diameter is 17 and its average path length is 8.4, almost double when compared to a social network like Facebook [39].

Local statistics Figure 1 illustrates the node degree distributions. In Figure 0a, the node degree is based on the number of connected nodes, with the distribution approximating a log-normal behavior. In Figure 0b, the node degree is based on the

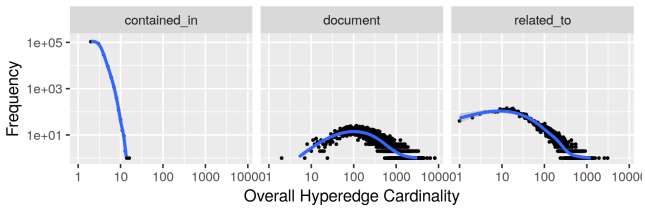
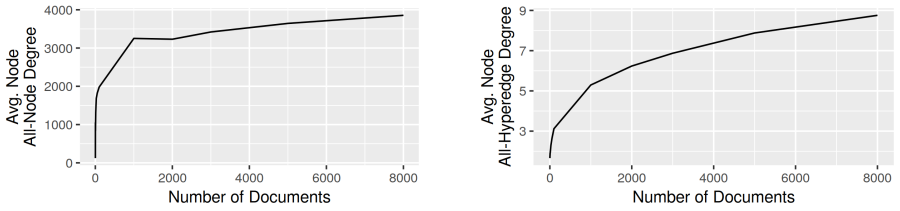


Figure 2: Hyperedge cardinality distribution based on the total number of nodes for the base model (log-log scale).



(a) Based on connected nodes.

(b) Based on connected hyperedges.

Figure 3: Average node degree over time for the base model.

number of connected hyperedges, with the distribution approximating a power law. This shows the usefulness of considering both of the node degrees in the hypergraph-of-entity, as they are able to provide different information.

Figure 2 illustrates the hyperedge cardinality distribution. For *document* hyperedges, cardinality is log-normally distributed, while for *related_to* hyperedges the behavior is slightly different, with low cardinalities having a higher frequency than they would in a log-normal distribution. Finally, the cardinality distribution of *contained_in* hyperedges, while still heavy-tailed, presents an initial linear behavior, followed by a power law behavior. The maximum cardinality for this type of hyperedge is also 16, which is a lot lower when compared to *document* hyperedges and *related_to* hyperedges, which have cardinality 8,167 and 3,084, respectively. This is explained by the fact that *contained_in* hyperedges establish a directed connection between a set of terms and an entity that contains those terms, being limited by the maximum number of words in an entity.

Temporal statistics In order to compute temporal statistics, we first generated 14 snapshots of the index based on a limit L of documents, for $L \in \{1, 2, 3, 4, 5, 10, 25, 50, 100, 1000, 2000, 3000, 5000, 8000\}$. Each snapshot was built based on the natural order of the documents found within the `tar.bz2` archives, up to a limit L, while the archives were accessed in directory order (i.e., the same as `ls -U` in Linux). This perfectly mimicked index growth, as documents were incrementally preprocessed and added to the hypergraph-of-entity.

Figure 3 illustrates the node-based and hyperedge-based average node degrees over time (represented as the number of documents in the index at a given instant). As we can see, both functions tend to converge, however this is clearer for the node-based degree, reaching nearly 4,000 nodes, through only 9 hyperedges, on average. Figure 4 illustrates the average undirected hyperedge cardinality over time, with a

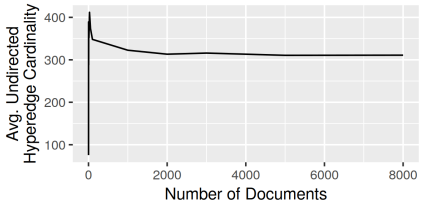


Figure 4: Average hyperedge cardinality over time for the base model.

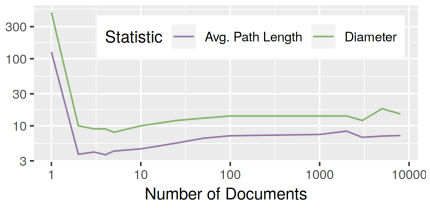


Figure 5: Average estimated diameter and average shortest path over time for the base model.

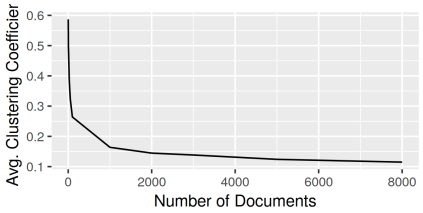


Figure 6: Average estimated clustering coefficient over time for the base model.

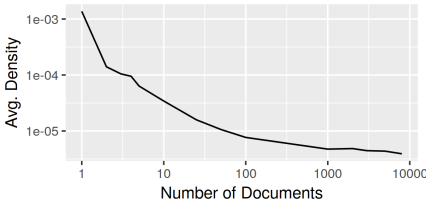


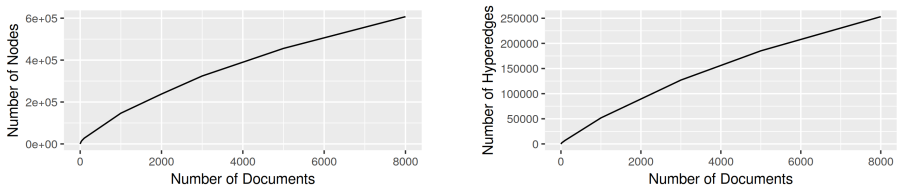
Figure 7: Average density over time for the base model.

convergence behavior that approximates 300 nodes per hyperedge, after rising to an average of 411.88 nodes for $L = 25$ documents.

Figure 5 illustrates the evolution of the average path length and the diameter of the hypergraph over time. For a single document, these values reached 126.1 and 491, respectively, while, for just two documents, they immediately lowered to 3.8 and 10. For higher values of L, both statistics increased slightly, reaching 7.2 and 15 for the maximum number of documents. Notice that these last values are equivalent to those computed in Table 2 (8.4 and 17, respectively), despite resulting in different amounts. This is due to the precision errors in our estimation approach, resulting in a difference of 1.2 and 2, respectively, which is tolerable when computation resources are limited. In Figure 6, we illustrate the evolution of the clustering coefficient, which rapidly decreases from 0.59 to 0.11. The low average path length and clustering coefficient point towards a weak community structure, possibly due to the coverage of diverse topics. However, we would require a random hypergraph generation model, like the Watts–Strogatz model [36] for graphs, in order to properly interpret the statistics.

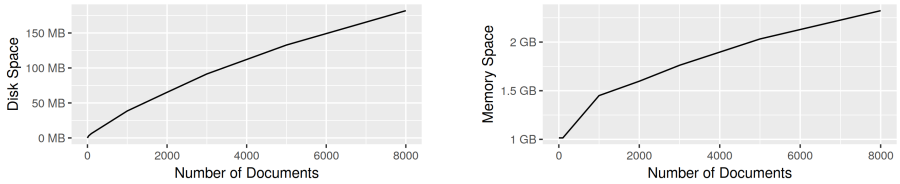
Figure 7 illustrates the evolution of the density over time. The density is consistently low, starting from 1.37e−03 and progressively decreasing to 3.91e−06 as the number of documents increases. This shows that the hypergraph-of-entity is an extremely sparse representation, with limited connectivity, which might benefit precision in a retrieval task.

Figure 8 displays the number of nodes (8a) and hyperedges (8b) created over time, as the index grew. Both presented a sub-linear growth behavior, reaching 4,566 nodes and 803 hyperedges for 10 documents, 238,141 nodes and 89,348 hyperedges for 2,000 documents, and 607,213 nodes and 253,154 for the whole collection of



(a) Nodes. (b) Hyperedges.

Figure 8: Number of nodes and hyperedges over time for the base model.



(a) Disk. (b) Memory.

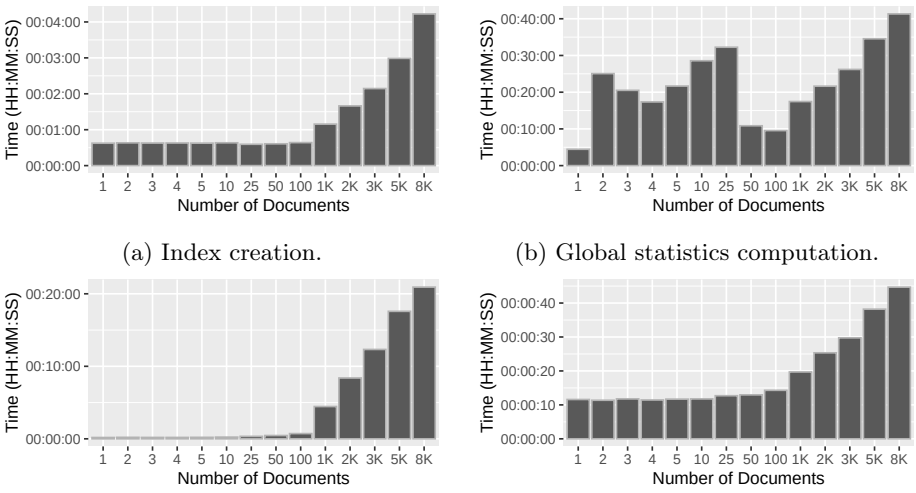
Figure 9: Required space for storing and loading the base model over time.

7,487 documents. The ratio of hyperedges per node evolved from 0.18, to 0.38, to 0.42, always staying below one. This means that the number of hyperedges increased slower than the number of nodes. Moreover, we know that nodes represent terms and entities, which will eventually converge to a finite vocabulary, further decreasing index growth rate.

As shown in Figure 9, we also measured the space usage of the hypergraph, both in disk (9a) and in memory (9b). In disk, the smallest snapshot required 43.8 KiB for one document, while the largest snapshot required 181.9 MiB for the whole subset. Average disk space over all snapshots was 37.5 MiB $\pm$ 58.9 MiB. In memory, for our particular application[6], the smallest snapshot used 1.0 GiB for one document, including the overhead of the data structures, and the largest snapshot used 2.3 GiB for the whole subset. Average memory space over all snapshots was 1.3 GiB $\pm$ 461.1 MiB. Memory also grew faster for the first 1,000 documents, apparently leading to an expected convergence, although we could not observe it for such a small subset.

Finally, Figure 10 illustrates the base model run times of the following operations for an increasing number of documents: index creation (10a); the computation of the global statistics (10b), also shown in Table 2; the computation of all node degrees (10c); and the computation of all hyperedge cardinalities (10d). As we can see, the most significant increase in run time happens around 1,000 documents, with the exception of the global statistics computation, which shows an increased run time for the first added documents. A possible reason for this anomaly is that this is the first analysis operation that we run after creating the index, which might influence the caching mechanisms of the system, thus reducing run time after the first documents and then resuming regular behavior. Indexing time took $1m09s$ for

[6] We relied on the Grph Java library, available at `http://www.i3s.unice.fr/~hogie/software/index.php?name=grph`, to represent the hypergraph in memory.



(a) Index creation.

(b) Global statistics computation.

(c) Node degrees computation.

(d) Hyperedge cardinalities computation.

Figure 10: Base model run time statistics.

1,000 documents and $4m13s$ for a maximum of 8,000 documents. The computation of global statistics took $17m26s$ for 1,000 documents and $41m18s$ for a maximum of 8,000 documents. Node degrees were computed in $4m27s$ for 1,000 documents, taking $20m55s$ at most, while hyperedge cardinalities were computed in only $19s$ for 1,000 documents, taking $44s$ at most, making it the most efficient statistic to compute.

6 Analyzing the structural impact of different index extensions

In this section, we extend our previous characterization work [11] by taking into consideration the index extensions, applied over the hypergraph-of-entity base model, as described by Devezas and Nunes [2, §4.1.2]. In Sections 6.1 and 6.2, we study the structural impact of synonyms and context, respectively. In Section 6.3, we propose a new grouping of terms based on the discretization of the term frequency (TF-bins), studying the structural impact of this index extension, while also considering different numbers of bins.

6.1 Synonyms

The base model for the hypergraph-of-entity establishes n-ary connections, both directed and undirected, among nodes that represent terms and entities. Most visibly, *document* hyperedges group all terms and entities mentioned in a document, a lot like a bag of words and entities that integrates both unstructured and structured evidence. This model can easily be extended with synonyms, that establish new bridges between documents. In particular, we used the synsets from WordNet 3.0 [40], based on the first sense of each term in the hypergraph, and only considering its noun form. Each synset was modeled as a *synonym* hyperedge. In this section, we characterize the hypergraph-of-entity when using the synonyms extension. We repeat the analysis described in Section 5, but only cover results that show a different behavior from the base model.

Table 3: Global statistics for the synonyms model.

Statistic	Value
Nodes	610,212
term	326,671
entity	283,541

Statistic	Value
Hyperedges	263,804
Undirected	25,588
document	7,484
related_to	7,454
synonym	10,650
Directed	238,216
contained_in	238,216

Statistic	Value
Avg. Degree	0.8646
Avg. Clustering Coefficient	0.1168
Avg. Path Length	7.5333
Diameter	17
Density	3.88e-06

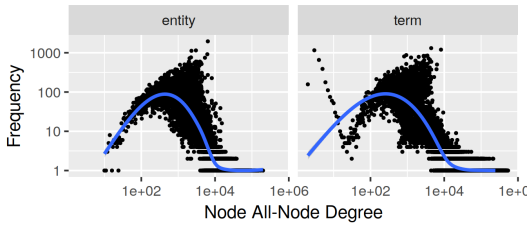


Figure 11: Node degree distribution, based on connected nodes, for the synonyms model (log-log scale).

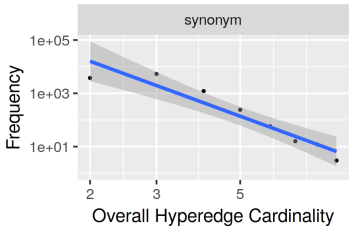


Figure 12: Synonym hyperedge cardinality distribution (log-log scale).

Table 3 shows the global statistics for the synonyms model. As we can see, the number of terms increased from 323,672 (cf. Table 2) to 326,671. This means that 2,999 synonym terms that did not originally belong to the collection were added. The number of undirected hyperedges increased significantly, with 10,650 new synonymy relations. The average degree slightly increased, with the average clustering coefficient and the density remaining stable. The diameter also remained at 17, however the average path length decreased almost a unit, from 8.37 to 7.53, approximating nodes through the relation of synonymy. This is an indicator of the usefulness of using synonyms to establish new bridges between documents. In fact, we found 4,558 new paths created by this extension, resulting in 65.29 documents linked on average per synonym. Besides global statistics, we also identified four interesting changes or new characteristics when compared to the base model:

- Term node degree distribution;
- Synonym hyperedge cardinality distribution;
- Average hyperedge cardinality over time;
- Average estimated diameter and average path length over time.

Term node degree distribution Figure 11 illustrates the node-based node degree distribution for entity and term nodes in the hypergraph-of-entity with the synonyms extension. While the behavior for entity nodes is similar to the base model, term nodes show a combination of a power law like behavior for the lower degrees, with a log-linear behavior for the remaining degrees. This is due to the introduction of synonyms from WordNet, which, as we can see in Figure 13, follow a distribution close to a power law.

Synonym hyperedge cardinality distribution Figure 12 illustrates the distribution of synonyms per hyperedge. As we can see, most *synonym* hyperedges either contain

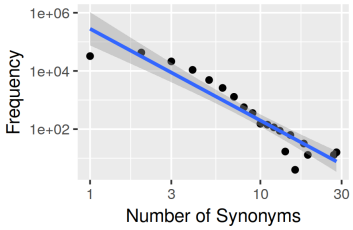


Figure 13: WordNet 3.0 noun synonyms distribution (log-log scale).

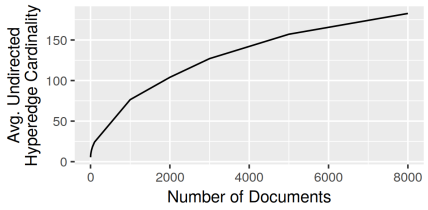


Figure 14: Average hyperedge cardinality over time for the synonyms model.

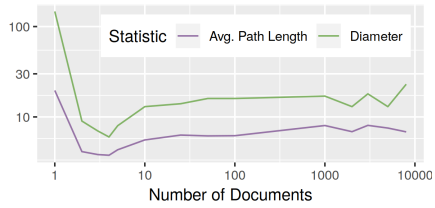


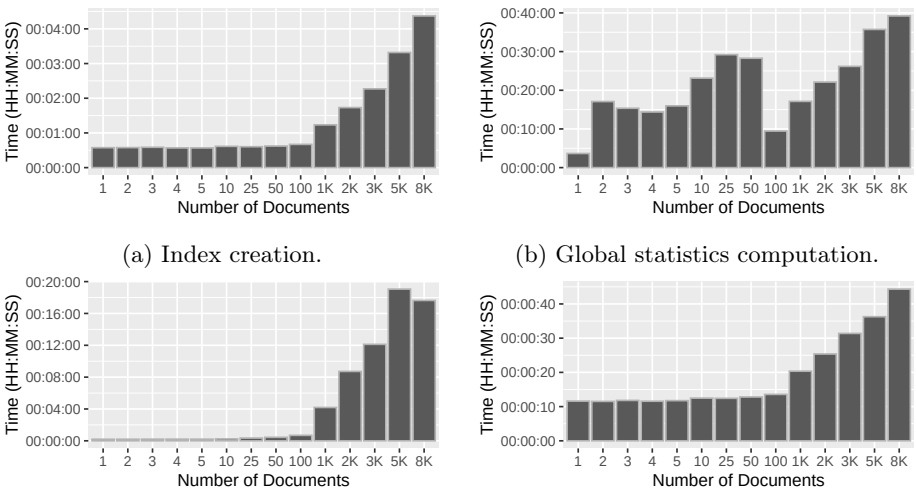
Figure 15: Average estimated diameter and average shortest path over time for the synonyms model.

two or three terms, while less than 100 hyperedges contain more than five synonyms. Most synonymy relations are ternary and, while there is not enough data to conclude it, the overall behavior approximates a power law.

Average hyperedge cardinality over time Consistent with the fact that most synsets introduced as undirected hyperedges have a low cardinality (two or three elements), the average hyperedge cardinality over time is overall lower than the base model. This is visible when comparing Figure 14 with Figure 4. Additionally, the behavior also changed from a fast growth and convergence behavior, in the base model, to a consistent sub-linear growth behavior. While convergence is not immediately clear in the synonyms model, the trend does point to such behavior.

Average estimated diameter and average path length over time With synonymy relations, both the average path length and the diameter start at a lower value than the base model, for only one document. Apart from the initial values, when comparing Figure 15 with Figure 5, we find a similar behavior, although the average path length decreases from 8.37, in the base model, to 7.53, in the synonyms model, when comparing a representation of the whole collection (cf. Tables 2 and 3). Despite the similar behavior, a unitary difference is quite significative in a network (e.g., in a social network like Facebook, the average path length is 4.74 [41], while in the original small-world study by Milgram [42, 43] the average path length was 6.2).

Temporal statistics of run times Finally, Figure 16 illustrates the synonyms model run times of the following operations for an increasing number of documents: index creation (16a); the computation of the global statistics (16b), also shown in Table 3;



(a) Index creation.

(b) Global statistics computation.

(c) Node degrees computation.

(d) Hyperedge cardinalities computation.

Figure 16: Synonyms model run time statistics.

the computation of all node degrees (16c); and the computation of all hyperedge cardinalities (16d). As we can see, similarly to what happened for the base model, the most significant increase in run time happens around 1,000 documents, with the exception of the global statistics computation, which shows an increased run time for the first added documents. We predict that the same caching mechanisms described for the base model are responsible for this anomaly. In Figure 16c, we also find a slight decrease in run time from 5,000 to 8,000 documents, which we do not find significant, as it was perhaps due to temporary load on the virtual machine. Indexing time took $1m13s$ for 1,000 documents and $4m22s$ for a maximum of 8,000 documents. The computation of global statistics took $17m07s$ for 1,000 documents and $39m13s$ for a maximum of 8,000 documents. Node degrees were computed in $4m11s$ for 1,000 documents, taking $19m03s$ at most, while hyperedge cardinalities were computed in only $20s$ for 1,000 documents, taking $44s$ at most, and maintaining the top rank in the most efficient statistic to compute, when compared to the base model.

6.2 Contextual similarity

Another way that we extended the base model was by using the contextual similarity between terms, as established based on the k-nearest neighbors according to word embeddings. For this particular analysis, word embeddings were obtained through word2vec, trained on a larger subset of the INEX 2009 Wikipedia collection, built from the documents mentioned in the relevance judgments for all 52 topics. The extracted vectors were of size 100, using sliding windows of 5 words to establish context, and ignoring words that appeared only once. Only the two nearest neighbors, with a similarity above 0.5 were considered to build the similarity graph. Contextual similarity hyperedges were then derived from this graph by iterating over each term and building sets that included the original term as well as incoming and outgoing terms.

Table 4: Global statistics for the contextual similarity model.

Statistic	Value
Nodes	697,068
term	413,527
entity	283,541

Statistic	Value
Hyperedges	410,371
Undirected	172,155
document	7,484
related_to	7,454
context	157,217
Directed	238,216
contained_in	238,216

Statistic	Value
Avg. Degree	1.1774
Avg. Clustering Coefficient	0.1423
Avg. Path Length	1.9333
Diameter	3
Density	2.75e-06

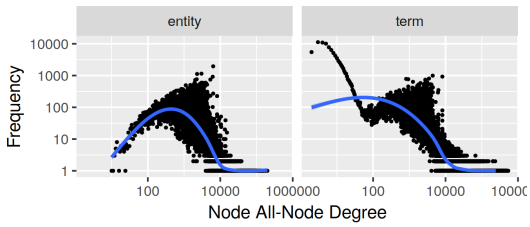


Figure 17: Node degree distribution, based on connected nodes, for the context model (log-log scale).

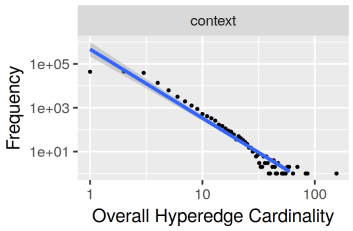


Figure 18: Context hyperedge cardinality distribution (log-log scale).

Table 4 shows the global statistics for the context model. As we can see, the number of terms significantly increased from 323,672 (cf. Table 2) to 413,527. This means that 89,855 contextually similar terms that did not originally belong to the collection were added — they were however a part of the larger 52 topics collection, otherwise no new terms would have been added. The number of undirected hyperedges also increased significantly, with 157,217 new context relations. The average degree also increased from 0.83 to 1.18, with the average clustering coefficient remaining stable and the density decreasing from $3.88e{-}06$ to $2.75e{-}06$. The diameter significantly decreased from 17 to 3, as did the average path length, which decreased from 8.37 to 1.93, strongly approximating nodes through the relation of contextual similarity. This is an indicator of the impact of using word embeddings to establish new bridges between documents, although we need to assess whether retrieval effectiveness will be affected by context as a kind of noise introduced in the process rather than a good discriminative feature. We found 42,145 new paths created by this extension, resulting in 23.03 documents linked on average per context. Notice that, although synonyms established a lower number of bridges, they also connected a higher number of documents on average (2.83× more than context). Only by studying retrieval effectiveness we will be able to assess which characteristic translates into a better performance in the model. Besides global statistics, we also identified four interesting changes or new characteristics when compared to the base model:

- Term node degree distribution;
- Context hyperedge cardinality distribution;
- Average hyperedge cardinality over time;
- Average estimated diameter and average path length over time;

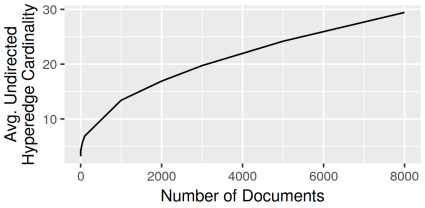


Figure 19: Average hyperedge cardinality over time for the context model.

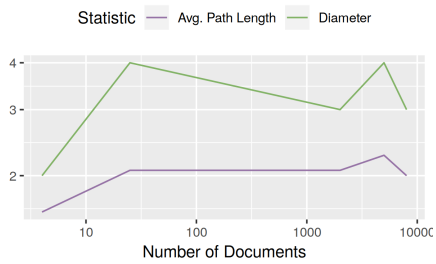


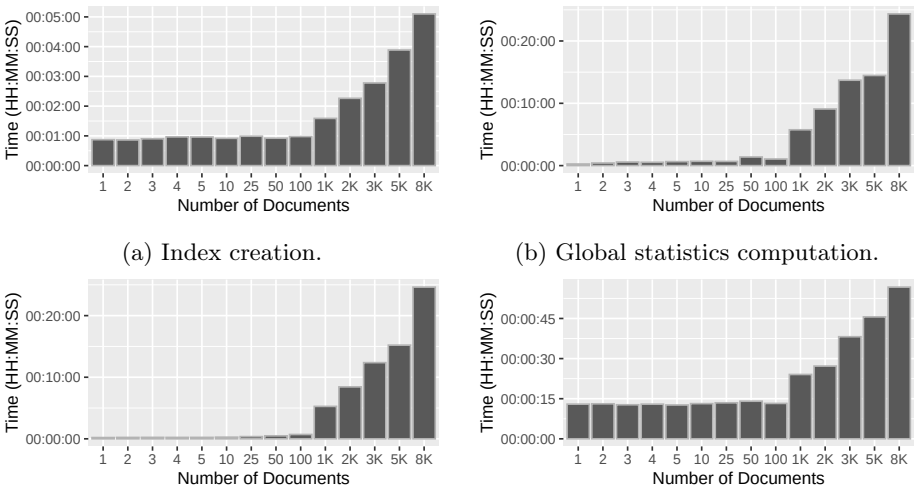
Figure 20: Average estimated diameter and average shortest path over time for the context model.

Term node degree distribution Figure 17 illustrates the node-based node degree distribution for entity and term nodes in the hypergraph-of-entity with the context extension. The behavior for entity nodes is similar to the base model and to the synonyms model. However, like in the synonyms model, term nodes show a combination of a power law like behavior for the lower degrees, with a log-linear behavior for the remaining degrees. Given the higher number of terms introduced through contextual similarity, we also find a distribution plot that is visually denser.

Context hyperedge cardinality distribution Figure 18 illustrates the distribution of terms per *context* hyperedge. As we can see, the behavior approximates a power law, with only a few *context* hyperedges containing around 50 nodes and one of them even reaching 156 nodes.

Average hyperedge cardinality over time Given the high number of introduced *context* hyperedges, most of them with a low cardinality, the average hyperedge cardinality was driven down, as we can see in Figure 19. In a similar way to the *synonym* hyperedges, the behavior also changed from a fast growth and convergence behavior, in the base model, to a consistent sub-linear growth behavior.

Average estimated diameter and average path length over time Perhaps one of the most interesting results of this analysis is the impact of index extensions in the diameter and average path length. This is particularly visible with the context extension — the diameter decreased from 17, in the base and similarity models, to only 3, in the context model. A similar behavior was identified for the average path length that decreased from 8.33 in the base model and 7.53 in the synonyms model, to only 1.93 in the context model. This behavior over time is seen in Figure 20, where, contrary to the base and synonyms model, we can find shorter geodesics immediately for a low number of documents. As an increasing part of the collection is considered, the length of the geodesics increase. This might be correlated with an increasing diversity of topics, thus being indicative of the discriminative power of the context extension, an aspect that should be further investigated in the future.



(a) Index creation.

(b) Global statistics computation.

(c) Node degrees computation.

(d) Hyperedge cardinalities computation.

Figure 21: Contextual similarity model run time statistics.

Temporal statistics of run times Finally, Figure 21 illustrates the contextual similarity model run times of the following operations for an increasing number of documents: index creation (21a); the computation of the global statistics (21b), also shown in Table 4; the computation of all node degrees (21c); and the computation of all hyperedge cardinalities (21d). As we can see, similarly to what happened for the base model, the most significant increase in run time happens around 1,000 documents. When compared to the base model and the synonyms model, the global statistics computation does not show an increased run time for the first added documents. This further supports the hypothesis of this being an anomaly that happened due to initial caching or load issue, particularly since the synonyms model is quite similar, structurally, to the context model. Indexing time took $1m35s$ for 1,000 documents and $5m05s$ for a maximum of 8,000 documents. The computation of global statistics took $5m44s$ for 1,000 documents and $24m20s$ for a maximum of 8,000 documents. Node degrees were computed in $5m15s$ for 1,000 documents, taking $24m37s$ at most, while hyperedge cardinalities were computed in only $24s$ for 1,000 documents, taking $56s$ at most, making it the most efficient statistic to compute, and maintaining the top rank in the most efficient statistic to compute, when compared to the base model and the synonyms model.

6.3 Term frequency bins

In this section, we analyze the TF-bins extension, which is based on the discretization of the term frequency per document. This way, term frequency can be added to the hypergraph-of-entity, while having a low impact in scalability (i.e., we remain focused on forming groups of nodes to minimize the space complexity of the representation model).

Table 5 shows the global statistics for the TF-bins model. As we can see, the number of nodes is the same as the original model, also remaining unchanged with the number of bins. The number of undirected hyperedges increased from 14,938

Table 5: Global statistics for the TF-bins model (bins=2 and bins=10).

Statistic	Bins	
	2	10
Nodes	607,213	607,213
term	323,672	323,672
entity	283,541	283,541

Statistic	Bins	
	2	10
Hyperedges	268,100	281,642
Undirected	29,884	43,426
document	7,484	7,484
related_to	7,454	7,454
tf_bin	14,946	28,488
Directed	238,216	238,216
contained_in	238,216	238,216

Statistic	Bins	
	2	10
Avg. Degree	0.8831	0.9277
Avg. Cl. Coef.	0.1021	0.1014
Avg. Path Len.	6.8333	6.9000
Diameter	13	14
Density	7.58e-06	7.86e-06

to 29,884 for two TF-bins, or to 43,426 with ten bins. The average degree slightly increased from 0.83 to 0.88 for two TF-bins per document, and then to 0.93 for ten TF-bins, with the average clustering coefficient remaining stable and the density increasing from $3.88e{-}06$ to $7.58e{-}06$ for two TF-bins, and then again slightly to $7.86e{-}06$ for ten TF-bins. The diameter decreased from 17 to 13 for two TF-bins, and 14 for ten TF-bins, as did the average path length, which decreased from 8.37 to 6.83 and 6.90 for two and ten TF-bins, respectively. When considering two TF-bins, we found 156,200 new paths created by this extension, resulting in 30.64 documents linked on average per TF-bin. When the number of bins increased to ten, the number of new paths decreased to 153,979, but the average number of documents linked per TF-bin increased to 37.99. Besides global statistics, we also identified seven interesting changes or new characteristics when compared to the base model:

- TF-bin hyperedge cardinality distribution per number of bins;
- Number of undirected hyperedges per number of bins;
- TF-bin hyperedges per number of bins;
- Diameter and average path length per number of bins;
- Average hyperedge cardinality over time per number of bins;
- Average density over time per number of bins.
- Average estimated diameter and average path length over time per number of bins;

Notice that, contrary to the synonyms and context extensions, the TF-bins extension did not affect the behavior of term node degree distribution, since it does not introduce external terms to the collection.

TF-bin hyperedge cardinality distribution Figure 22 illustrates the cardinality distribution of *tf_bin* hyperedges, for different numbers of bins. The behavior is similar to the *related_to* hyperedges, however, as the number of bins increases, lower values of cardinality become more frequent and the behavior starts tending towards a power law.

Number of hyperedges per number of bins As expected, in Figure 23a, we find a growth in the number of undirected hyperedges, from 29,884, for two bins, to 43,426, for ten bins. The same happens for the *tf_bin* hyperedges (Figure 23b), which are responsible for propelling such growth. The amount of hyperedges generated by increased TF-bins will eventually converge, since there is a limited number of terms per document to segment. However, for this collection, it is clear that the number of TF-bins can range from two to ten, while always generating new hyperedges, increasing the granularity at which term frequency will contribute to the model.

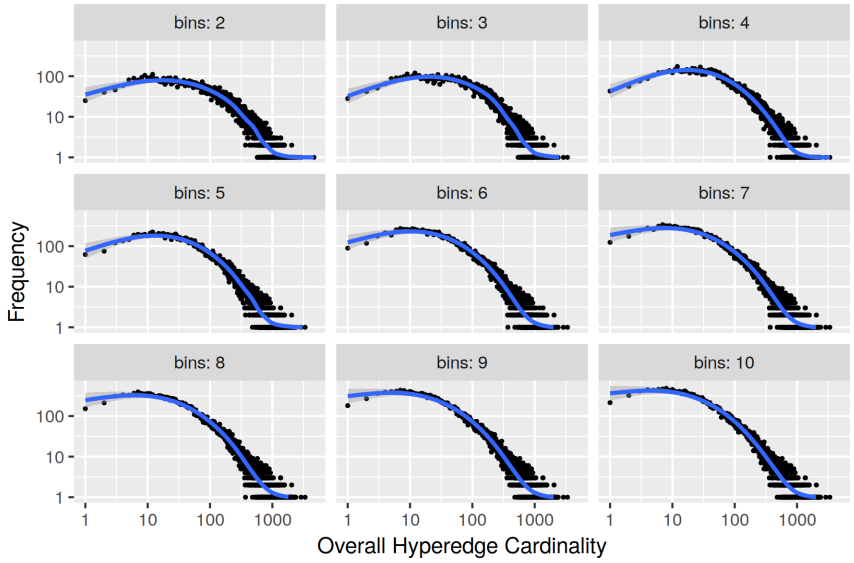
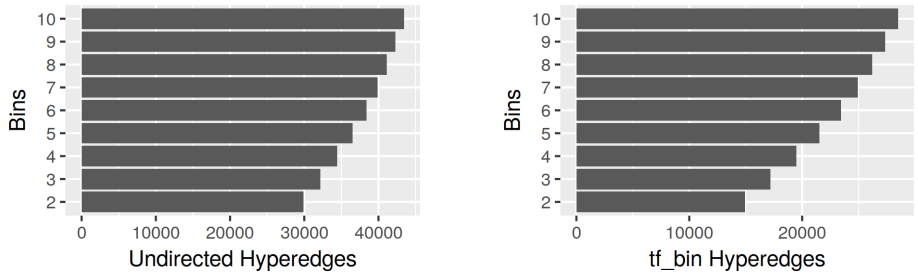


Figure 22: TF-bin hyperedge cardinality distribution (log-log scale).



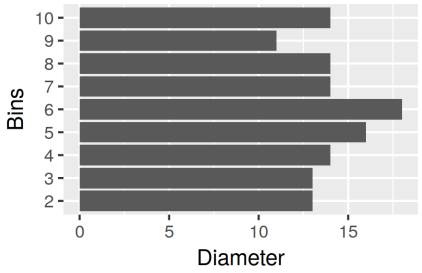
(a) Undirected hyperedges.

(b) TF-bin hyperedges.

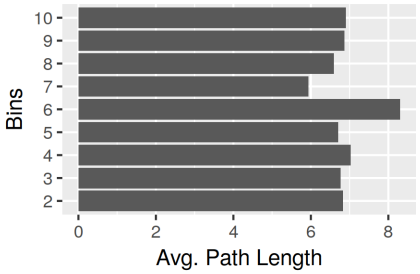
Figure 23: Number of hyperedges, per number of bins, for the TF-bins model.

Diameter and average path length per number of bins As show in Figure 24, both the diameter and the average path length, which correspond to the maximum and average geodesic distances in the hypergraph, show a high variability with the number of bins. In particular, the diameter and average path length both reach their maximum values of 18 and 8.30 when using 6 TF-bins. The minimum diameter of 11 is reached when using 9 TF-bins, while the minimum average path length of 5.93 is reached when using 7 TF-bins. This suggests that the number of bins might influence retrieval effectiveness, if varying the diameter and the average path length also affects performance directly.

Average hyperedge cardinality over time Figure 26 shows the evolution of the average hyperedge cardinality for different numbers of bins. The behavior is similar to the base model (cf. Figure 4), which is equivalent to having one TF-bin. As the number of TF-bins increases, the overall average hyperedge cardinality decreases, which is the expected behavior. This is less visible as the number of bins reaches



(a) Diameter.



(b) Average path length.

Figure 24: Geodesic-based metrics, per number of bins, for the TF-bins model.

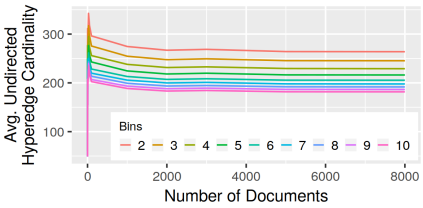


Figure 25: Average hyperedge cardinality over time, per number of bins, for the TF-bins model.

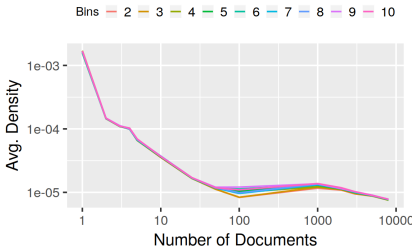


Figure 26: Average density over time, per number of bins, for the TF-bins model.

a higher value, at which point the overall cardinality is less affected, showing a progressively lower decreasing behavior. While the number of TF-bins affects this characteristic of the hypergraph, the overall behavior is maintained.

Average density over time The average density shown in Figure 26 follows a similar behavior to the base model (cf. Figure 7), regardless of the number of TF-bins. However, there is a small variation for the interval of approximately 100 to 1,000 documents, after which it is once again reduced to the same value for the different numbers of TF-bins. It is perhaps the diversity in term frequency introduced for documents in this interval that promotes such a difference. This would explain the creation of a higher number of *tf_ bin* hyperedges, without empty TF intervals (e.g., $]2, 2]$).

Average estimated diameter and average shortest path over time Figure 27 shows the evolution of the diameter and average path length, over an increasing number of documents and TF-bins. Apart from both metrics reaching higher values for a single document as well as for five TF-bins, the behavior is similar to the base model (cf. Figure 5).

Temporal statistics of run times Finally, Figures 28 and 29 illustrate the TF-bins model run times of the following operations for an increasing number of documents:

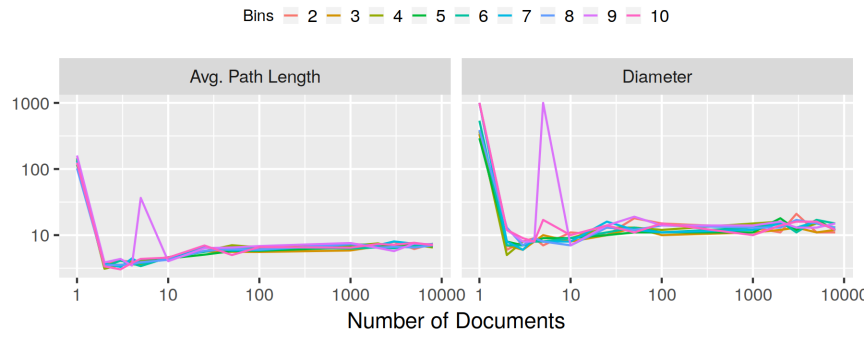
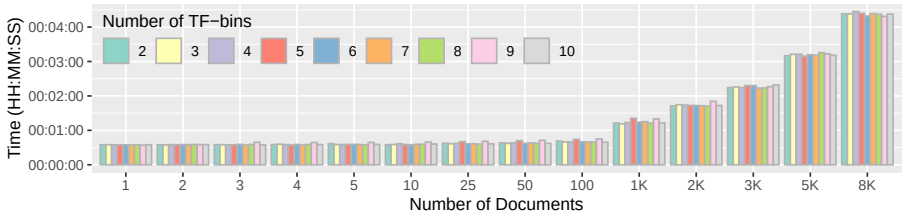
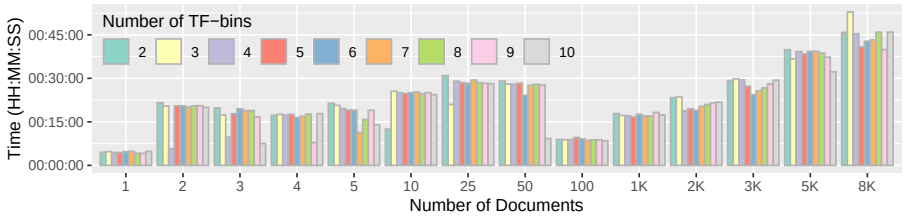


Figure 27: Average estimated diameter and average shortest path over time, per number of bins, for the TF-bins model.



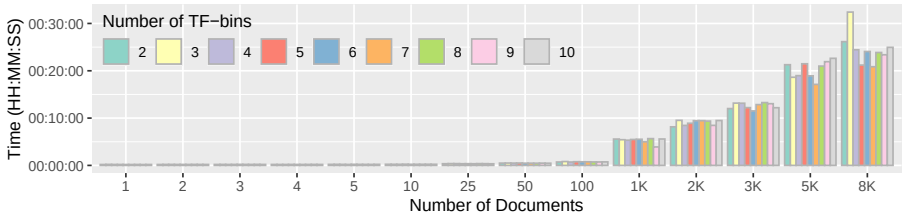
(a) Index creation.



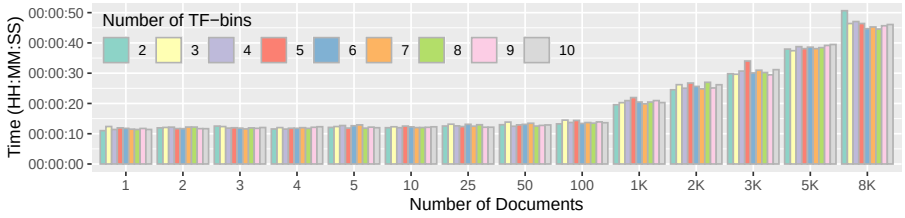
(b) Global statistics computation.

Figure 28: TF-bins models run time statistics (part 1).

index creation (28a); the computation of the global statistics (28b), also shown in Table 5; the computation of all node degrees (29a); and the computation of all hyperedge cardinalities (29b). As we can see, similarly to what happened for the base model and the synonyms model, the most significant increase in run time happens around 1,000 documents, with the exception of the global statistics computation, which shows an increased run time for the first added documents. Indexing time took $1m11s$ for 1,000 documents and $4m27s$ for a maximum of 8,000 documents. The computation of global statistics took $16m38s$ for 1,000 documents and $52m50s$ for a maximum of 8,000 documents. Node degrees were computed in $3m54s$ for 1,000 documents, taking $32m23$ at most, while hyperedge cardinalities were computed in only $19s$ for 1,000 documents, taking $50s$ at most, making it the most efficient statistic to compute, maintaining the top rank in the most efficient statistic to compute, in line with the other studied models models.



(a) Node degrees computation.



(b) Hyperedge cardinalities computation.

Figure 29: TF-bins models run time statistics (part 2).

Table 6: Evaluating the different models in the ad hoc document retrieval task.

Model	MAP	NDCG@10	P@10	MAP	NDCG@10	P@10	MAP	NDCG@10	P@10
Lucene TF-IDF	0.2160	0.2667	0.2800	0.2160	0.2667	0.2800	0.2160	0.2667	0.2800
Lucene BM25	0.3412	0.5479	0.4900	0.3412	0.5479	0.4900	0.3412	0.5479	0.4900
HGoE RWS		$\ell = 1$			$\ell = 2$			$\ell = 3$	
Base model	0.0046	0.0799	0.0400	0.0039	0.0718	0.0400	0.0028	0.0576	0.0400
Synonyms	0.0013	0.0440	0.0200	0.0024	0.0799	0.0400	0.0023	0.0718	0.0400
Context	0.0000	0.0000	0.0000	0.0010	0.0220	0.0100	0.0010	0.0220	0.0100
TF-bins$_2$	**0.1082**	**0.2443**	0.2100	0.1025	0.1730	**0.2000**	0.0918	0.1302	0.1400
TF-bins$_3$	0.0911	0.2004	0.2200	0.0989	0.0954	0.1200	0.0868	0.0751	0.1000
TF-bins$_4$	0.0957	0.1969	0.2000	0.1107	**0.2007**	0.1900	0.0928	0.1669	0.1700
TF-bins$_5$	0.1049	0.2355	0.2400	0.1050	0.1364	0.1400	0.0954	0.1121	0.1400
TF-bins$_6$	0.1057	0.2405	**0.2600**	0.1108	0.1906	**0.2000**	0.1022	**0.1792**	**0.1900**
TF-bins$_7$	0.1000	0.2212	0.2500	0.1072	0.1255	0.1200	0.0939	0.0934	0.1000
TF-bins$_8$	0.0894	0.2131	0.2100	0.1078	0.0988	0.1100	0.0966	0.0641	0.0800
TF-bins$_9$	0.0954	0.1494	0.1500	0.1107	0.1402	0.1500	0.0958	0.1069	0.1200
TF-bins$_{10}$	0.1062	0.2127	0.2200	**0.1133**	0.1436	0.1600	**0.1079**	0.1143	0.1300

7 An application to information retrieval

So far, we have analyzed the structural impact of different index extensions in regards to the characteristics of the hypergraph. However, there is little value in understanding the behavior of structural features without the context of its application, which in this case is in the area of information retrieval [2]. Thus, we assess the effectiveness of each model, with different extensions and parameter configurations, through a classical information retrieval evaluation process, based on the 10 topic subset of the INEX 2009 Wikipedia collection (INEX 2009 10T-NL).

We launched three evaluation runs per index configuration, i.e., for different versions of the HGoE (hypergraph-of-entity) representation model based on different extensions. We relied on the RWS ranking function, experimenting with different random walk lengths $\ell \in \{1,2,3\}$, and a fixed configuration for the remaining parameters: $r = 10{,}000$, *expansion* disabled (i.e., without seed node selection [2, §4.2.1]), and *weights* enabled (i.e., considering *tf_bin* hyperedge weights, the only available weights in the indexes).

Table 6 shows the MAP (mean average precision), NDCG@p (normalized discounted cumulative gain at a cutoff of p), and P@n (precision at a cutoff of n),

Table 7: Comparing the global statistics for the different models.

Model	Nodes	Hyperedges	Degree	Cl. Coef.	Avg. Path Len.	Diam.	Density
Base model	607,213	253,154	0.8338	0.1148	8.3667	17	3.88e-06
Synonyms	610,212	263,804	0.8646	0.1168	7.5333	17	3.88e-06
Context	697,068	410,371	1.1774	0.1423	1.9333	3	2.75e-06
TF-bins$_2$	607,213	268,100	0.8831	0.1021	6.8333	13	7.58e-06
TF-bins$_3$	607,213	270,359	0.8905	0.1011	6.7667	13	7.65e-06
TF-bins$_4$	607,213	272,649	0.8980	0.0999	7.0333	14	7.60e-06
TF-bins$_5$	607,213	274,698	0.9048	0.0996	6.7000	16	7.73e-06
TF-bins$_6$	607,213	276,615	0.9111	0.1029	8.3000	18	7.69e-06
TF-bins$_7$	607,213	278,087	0.9159	0.1010	5.9333	14	7.82e-06
TF-bins$_8$	607,213	279,356	0.9201	0.1034	6.6000	14	7.83e-06
TF-bins$_9$	607,213	280,524	0.9240	0.0994	6.8667	11	7.84e-06
TF-bins$_{10}$	607,213	281,642	0.9277	0.1014	6.9000	14	7.86e-06

computed for the relevance judgments provided by the INEX 2010 Ad Hoc track [44]. As we can see, by analyzing the maximum values per column (in bold), the TF-bin models were able to obtain significantly better results overall, when compared to the base model, the synonyms model, and the context model. None of the HGoE models is yet able to outperform the baselines, although TF-bins are able to approximate TF-IDF in regard to NDCG@10 and P@10. The hypergraph-based models need to be reiterated over and improved. Herein lies the usefulness of computing the properties of the hypergraph structures and analyzing the hypergraph-of-entity. While there is no clear pattern of effectiveness correlated with the number of bins, if we consider the NDCG@10 scores, the best model for $\ell = 1$ is TF-bins$_2$, the best model for $\ell = 2$ is TF-bins$_4$, and the best model for $\ell = 3$ is TF-bins$_6$. This might indicate that a higher number of bins works best with a longer random walk length. However, there is no concordance to support this hypothesis when looking at the MAP and P@10 metrics, thus further investigation is required.

In order to better understand whether there is a direct relation between any of the computed structural features of the hypergraph and the effectiveness of the retrieval model, we first summarize the structural features for each model in Table 7. By comparing each feature with the evaluation metrics from Table 6, we are able to find some indicators of (in)effectiveness in a graph-based retrieval model. According to Table 6, context was the worst performing model, over all values of ℓ. The context model also has the highest average degree and clustering coefficient, as well as the lowest average path length and diameter (cf. Table 7). This indicates that a higher local connectivity and an overall lower distance between nodes might not beneficial for retrieval effectiveness. We also observe that the TF-bin models, which have the best performance, also have a lower clustering coefficient than the base, synonyms and context models, ranging between 0.0994 and 0.1034.

We also studied the structural impact of each extension, through the relative change to individual features, in comparison to the base model. Figure 30 shows a heatmap based on the change percentages in regards to the base model, which, by definition, has a 0% change over all features, in comparison to itself. As we can see, the context model suffered the most evident overall change, with a -467% change in diameter, and a -333% change in average path length. This model is of particular interest, as it resulted in the worst retrieval performance, when compared to the remaining models. Interestingly, this is also visible in its structural features.

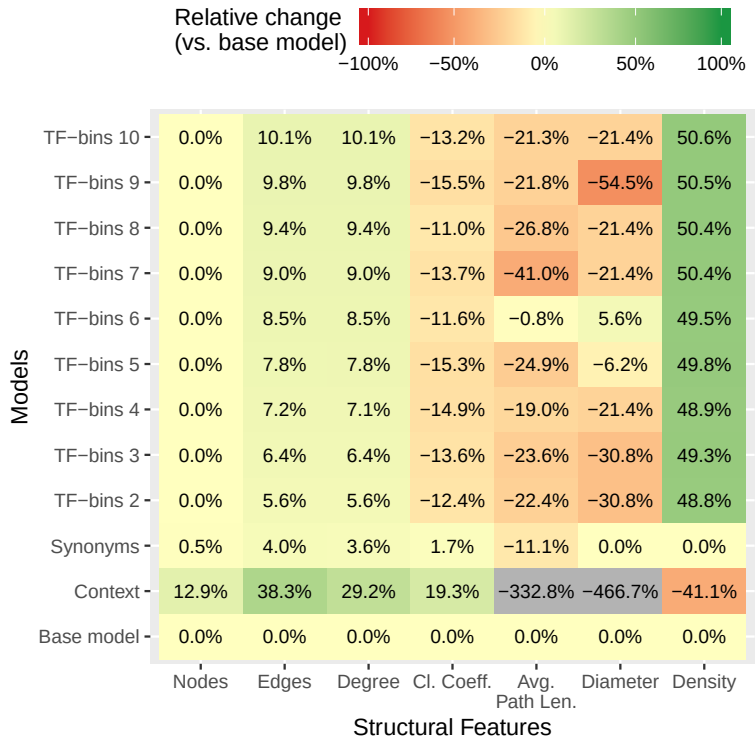


Figure 30: Relative change of structural features when compared to the base model.

The clustering coefficient for the context model also suffered a substantial increase in relation to the base model, with a change of 19%, as did the degree, with a change of 29%. When looking at the density for all models, there was no change for the synonyms model, but there was a positive change, rounding 50% (in green), for the TF-bins models, and there was a negative change of −41% for the context model. The number of nodes suffered no change for the TF-bins models, but there a slight increase for synonyms (as new terms from synsets were added), and a more significative increase for the context model. The number of edges suffered a consistently larger increase for TF-bins models, as the number of bins increased, with the synonyms model showing a slight increase, and the context model once again showing a more significative increase.

7.1 Correlating evaluation metrics and structural features

In Table 8 we further organize this approach, by comparing the evaluation results of each metric with the values of each structural feature. By using Spearman's rank correlation coefficient (ρ), we can verify whether the retrieval model's performance ranking given by the evaluation metrics (our ground truth) can compare with the ranking given by any of the structural features, as computed for each model. Let us first follow up with the indicators we put forth in the manual comparison of the two tables.

We proposed that a high average degree and clustering coefficient would result in a low MAP, NDCG@10 and P@10, which does not necessarily mean that either

Table 8: Spearman's ρ between evaluation metrics and structural features.

		Nodes	Hyperedges	Degree	Cl. Coef.	Avg. Path Len.	Diam.	Density
$\ell = 1$	**MAP**	-0.6504	0.0559	0.0559	-0.5245	0.0979	0.1000	0.5009
	NDCG@10	-0.6504	-0.0350	-0.0350	-0.3636	-0.1119	0.2000	0.4308
	P@10	-0.6527	0.1018	0.1018	-0.4667	-0.0982	0.3047	0.5800
$\ell = 2$	**MAP**	-0.6516	0.4098	0.4098	-0.5464	0.2172	0.1449	0.8035
	NDCG@10	-0.5913	0.0699	0.0699	-0.5804	0.2797	0.1036	0.4448
	P@10	-0.6242	0.0035	0.0035	-0.5519	0.2882	0.0593	0.4049
$\ell = 3$	**MAP**	-0.6504	0.4615	0.4615	-0.4685	0.0699	0.1965	0.8932
	NDCG@10	-0.5322	-0.0280	-0.0280	-0.5524	0.3357	0.2000	0.3573
	P@10	-0.6242	-0.0211	-0.0211	-0.6151	0.2707	0.1993	0.3873

feature is a good overall discriminator of model performance. In fact, the average degree does not show correlation consistency among the different evaluation metrics and parameter configurations. On the other hand, the clustering coefficient is negatively correlated with each evaluation metric over the different random walk length parameter configurations, ranging between -0.61 and -0.36. This makes the clustering coefficient a weak, but consistent indicator of the performance of graph-based retrieval models (i.e., higher values of the clustering coefficient indicate a low retrieval effectiveness). Absolute correlation is not particularly high, since retrieval performance does not solely depend on the structure of the graph, but also on the semantics of the representation model.

We also proposed that a low average path length and diameter would be indicative of low model performance. While the average path length and diameter correlations with the evaluation metrics are mostly positive, these are not sufficiently consistent to be considered good global indicators of performance. There are, however, special cases when the average path length serves as a slight indicator of performance, namely for $\ell > 1$ and for the top 10 results. For $\ell = 1$, there is a slight negative correlation that could be explained by the fact that this model only relies on the immediate neighborhood within the hypergraph and does not depend on short paths for connectivity. The diameter, on the other side, always shows a positive correlation with the evaluation metrics, but its absolute value is overall low and inconsistent for it to provide a good discriminative indicator of retrieval performance.

With a similar behavior to the clustering coefficient, but with an inverse sign, the density was overlooked as a good indicator of model performance. In particular, the worst performing model (context model) also has the lowest density of $2.75e{-}06$, followed by the base model and the synonyms model, tied at a density of $3.88e{-}06$, and then by the TF-bin models, with densities ranging from $7.58e{-}06$ to $7.86e{-}06$. While the density is a good discriminative of graph-based retrieval models, its granularity is low, only properly distinguishing between models with an obvious difference in performance.

7.2 Design rules for modifying or extending the hypergraph-of-entity

After the analysis of the impact of structural features in the performance of the retrieval models, we reflect on the implications of our findings. We use these findings to prepare a set of rules that serve as indicators or as a guide for the continued redesign of the hypergraph-of-entity. In particular, the guidelines we propose should be helpful in the process of comparing different versions based on modifications or extensions to our model. We propose two classes of indicators:

Table 9: Indicators of graph-based retrieval model performance.

Ranking indicators		Anomaly indicators	
Cl. Coef.	Ascending order $\sim 50\%$ correlated with retrieval performance.	Degree	Abnormally high values ($> \mu + 2\sigma$) indicate a low performing model.
Density	Descending order $\sim 50\%$ correlated with retrieval performance.	Diameter	Abnormally low values ($< \mu - 2\sigma$) indicate a low performing model.

Ranking indicators Structural features that can be used to rank different graph-based models in regards to their predicted retrieval performance.

Anomaly indicators Structural features that cannot be used to rank graph-based models based on retrieval performance, but can, however, be useful for identifying anomalous models with a high chance of a low performance.

Table 9 shows the identified ranking and anomaly indicators according to the analysis carried at the beginning of this section. The clustering coefficient and the density were both identified as ranking indicators with an approximate certainty rate of 50%, based on an ascending and descending order, respectively. The degree and diameter were identified as anomaly indicators, with the degree being used to identify abnormally high values, for example larger than two standard deviations (2σ) above the mean (μ), and the diameter being used to identify abnormally low values, for example less than two standard deviations below the mean.

8 Conclusion

We characterized the hypergraph-of-entity representation model, based on the structural features of the hypergraph. We analyzed the node degree distributions, based on nodes and hyperedges, and the hyperedge cardinality distributions, illustrating their distinctive behavior. We also analyzed the temporal behavior, as documents were added to the index, studying average node degree and hyperedge cardinality, estimated average path length, diameter and clustering coefficient, as well as density and space usage requirements. We expanded on the characterization work by analyzing different model extensions based on synonymy, contextual similarity, and a new concept of TF-bins, and we also measured the run time of several operations like indexing and the computation of properties. Our contributions included the application of two strategies for the approximation of statistics based on the shortest distance, as well as the clustering coefficient. We also proposed a simple approach for computing the density of a general mixed hypergraph, based on an induced bipartite mixed graph. Finally, we focused on the application of this characterization work, which, we proposed, should inform the design of graph-based representation models for information retrieval. In particular, we studied the change in structural features, when compared to the base model, as well as the correlations between retrieval effectiveness metrics (MAP, NDCG@10, P@10) and structural features (e.g., average degree, clustering coefficient). While structural features rarely presented a higher than 50% absolute correlation with any of the evaluation metrics, we identified some of them as indicators useful for ranking the retrieval models according to their effectiveness, or for identifying anomalies that lead to low effectiveness. More importantly, we have provided an analysis framework for hypergraphs that can easily be implemented and applied to both small and large-scale hypergraphs. We have

also provided a characterization based on this framework, illustrating the behavior of several statistics, for instance showing that, while the degree distribution based on hyperedges still follows a power law, like in real-world networks represented as graphs, the degree distribution based on nodes instead approximates a log-normal distribution. During the development of this work, we have also found that:

- Few attention has been given to hypergraph characterization in the real-world;
- The community is still lacking in tools to analyze hypergraphs:
  - There is no *de facto* library for hypergraph analysis;
  - Few file formats support hypergraphs, namely with directed hyperedges.
- Polyadism introduces additional complexity and calls for novel metrics that take the information within collective relations into account.

Future work In the future, we would like to further explore the computation of density, since the bipartite-based density we proposed, although useful, only accounts for hyperedges already in the hypergraph. We would also like to study the parameterization of the two estimation approaches we proposed, based on random walks and node sampling. Despite their straightforward definition, these approaches also require further evaluation, in order to understand what the expected error will be for different configurations. Another open challenge is the definition of random hypergraph generation model, which would be useful to improve characterization. Additionally, several opportunities exist in the study of the hypergraph at a mesoscale, be it identifying communities, network motifs or graphlet, or exploring unique patterns to hypergraphs. It would also be interesting to include centrality metrics in the correlation analysis, in order to understand for instance whether closeness or betweenness might impact retrieval effectiveness in the hypergraph-of-entity, furthermore considering multiple combinations of extensions, as opposed to a single one, as we have done here. Finally, regarding the hypergraph-of-entity model, it would also be useful to repeat the analysis we describe in this work based on additional test collections, as to support or disprove the results we found. Perhaps future TREC or CLEF tracks could provide relevance judgments for multiple tasks in entity-oriented search, which would be useful to boost the study of generality in information retrieval.

List of abbreviations

HGoE	hypergraph-of-entity
INEX	INitiative for the Evaluation of XML Retrieval
MAP	mean average precision
NDCG@p	normalized discounted cumulative gain at a cutoff of p
OWL	web ontology language
P@n	precision at a cutoff of n
qrels	query relevance set
RDF	resource description framework
RWS	random walk score
TF	term frequency
TF-bin	term frequency bin
TF-IDF	term frequency $\times$ inverted document frequency
YAGO	Yet Another Great Ontology

Availability of data and materials

The INEX 2009 Wikipedia collection analysed during the current study is available at the Max-Planck-Institut für Informatik website for the Databases and Information Systems department, `https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/software/inex/`. The topics and relevance judgments for the INEX 2010 Ad Hoc track are available at the INEX website, `http://inex.mmci.uni-saarland.de/data/documentcollection.html`.

The remaining datasets that were generated and analysed during the current study are available from the corresponding author on reasonable request.

The software required to replicate this study is available with the name Army ANT, under the BSD 3-Clause license, at `https://github.com/feup-infolab/army-ant`.

Competing interests
The authors declare that they have no competing interests.

Funding
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.

Authors' contributions
JLD and SSN have jointly discussed and developed the ideas present in this work. JLD was responsible for the data processing and analysis, and for writing the manuscript. JLD and SSN jointly reviewed the manuscript, with SSN being the main contributor to this process, promoting discussion that led to the heatmap depicting the relative change of structural features, which was prepared by JLD.

Acknowledgements
We would like to thank Bruno Martins, from INESC-ID and the University of Lisbon, for his suggestion on integrating the concept of term frequency into the hypergraph-of-entity in the form of bins.

Authors' information
JLD is a doctoral student at MAP-i (`https://mapi.map.edu.pt/`), the Doctoral Program in Computer Science of the Universities of Minho, Aveiro, and Porto. He is affiliated with FEUP InfoLab and INESC TEC. His ongoing thesis, entitled "Graph-Based Entity-Oriented Search", was born from his recurrent fascination with connecting data and building general models to help people solve their information needs. He has done work in several domains, including information retrieval, music recommender systems, network science, and data visualization. He is currently exploring the usage of hypergraphs as a joint representation for corpora and knowledge bases, with the goal of proposing a universal ranking function for entity-oriented search, while improving retrieval effectiveness. More information can be found at `http://josedevezas.com/`.

SSN is an Assistant Professor at the Department of Informatics Engineering at the Faculty of Engineering of the University of Porto (FEUP), and a Senior Researcher at the Centre for Information Systems and Computer Graphics at INESC TEC. He holds a Ph.D. in Information Retrieval (2010) focused on using temporal features for relevance estimation, and an MSc in Information Management. His research interests are in the areas of Information Retrieval and Web Information Systems, in particular in the use of temporal features for ranking, the study of information dynamics on the Web, and Computational Journalism. More information and selected publications can be found at `https://web.fe.up.pt/~ssn/`.

References

1. Estrada, E., Rodriguez-Velazquez, J.A.: Complex networks as hypergraphs. arXiv preprint physics/0505137 (2005)
2. Devezas, J., Nunes, S.: Hypergraph-of-entity: A unified representation model for the retrieval of text and knowledge. Open Computer Science **9**(1), 103–127 (2019). doi:10.1515/comp-2019-0006
3. Bast, H., Buchhold, B., Haussmann, E., *et al.*: Semantic search on text and knowledge bases. Foundations and Trends® in Information Retrieval **10**(2-3), 119–271 (2016)
4. Csardi, G., Nepusz, T., *et al.*: The igraph software package for complex network research. InterJournal, Complex Systems **1695**(5), 1–9 (2006)
5. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An open source software for exploring and manipulating networks. In: Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009 (2009). http://aaai.org/ocs/index.php/ICWSM/09/paper/view/154
6. Himsolt, M.: GML: A portable graph file format. Technical report, Universität Passau (1997)
7. Brandes, U., Eiglsperger, M., Herman, I., Himsolt, M., Marshall, M.S.: Graphml progress report structural layer proposal. In: International Symposium on Graph Drawing, pp. 501–512 (2001). Springer
8. Ouvrard, X., Goff, J.L., Marchand-Maillet, S.: Adjacency and tensor representation in general hypergraphs part 1: e-adjacency tensor uniformisation using homogeneous polynomials. CoRR **abs/1712.08189** (2017). 1712.08189
9. Aparicio, D., Ribeiro, P., Silva, F.: Graphlet-orbit transitions (got): A fingerprint for temporal network comparison. PLoS One **13**, 0205497 (2018). doi:10.1371/journal.pone.0205497
10. Berge, C.: Graphes et Hypergraphes. Monographies universitaires de mathematiques. Dunod, Paris (1970)
11. Devezas, J.L., Nunes, S.: Characterizing the hypergraph-of-entity representation model. In: Complex Networks and Their Applications VIII - Volume 2 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019, Lisbon, Portugal, December 10-12, 2019, pp. 3–14 (2019). doi:10.1007/978-3-030-36683-4_1
12. Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., Petrelli, D.: Hybrid search: Effectively combining keywords and semantic searches. In: European Semantic Web Conference, pp. 554–568 (2008). Springer
13. Bast, H., Buchhold, B.: An Index for Efficient Semantic Full-text Search. In: Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management, pp. 369–378 (2013). doi:10.1145/2505515.2505689

14. Voorhees, E.M.: The efficiency of inverted index and cluster searches. In: SIGIR'86, Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, September 8-10, 1986, pp. 164–174 (1986). doi:10.1145/253168.253203
15. Zobel, J., Moffat, A., Ramamohanarao, K.: Inverted files versus signature files for text indexing. ACM Trans. Database Syst. **23**(4), 453–490 (1998). doi:10.1145/296854.277632
16. Halpin, H.: A query-driven characterization of linked data. In: Proceedings of the WWW2009 Workshop on Linked Data on the Web, LDOW 2009, Madrid, Spain, April 20, 2009. (2009)
17. Ge, W., Chen, J., Hu, W., Qu, Y.: Object link structure in the semantic web. 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, pp. 257–271 (2010). doi:10.1007/978-3-642-13489-0_18
18. Fernández, J.D., Martínez-Prieto, M.A., de la Fuente Redondo, P., Gutiérrez, C.: Characterizing rdf datasets. Journal of Information Science **1**, 1–27 (2016)
19. Erdös, P.: On some extremal problems on r-graphs. Discret. Math. **1**(1), 1–6 (1971). doi:10.1016/0012-365X(71)90002-1
20. Brown, W., Erdos, P., Sós, V.: Some extremal problems on r-graphs. In: New Directions in the Theory of Graphs (Proc. Third Ann Arbor Conf., Univ. Michigan, Ann Arbor, Mich, 1971), pp. 53–63 (1973)
21. Sperner, E.: Ein satz über untermengen einer endlichen menge. Mathematische Zeitschrift **27**(1), 544–548 (1928)
22. Turán, P.: On an extremal problem in graph theory. Matematikai és Fizikai Lapok **48**, 436–452 (1941)
23. Turán, P.: Research problems. Magyar Tud. Akad. Mat. Kutato Internat. Közl. **6**, 417–423 (1961)
24. Erdös, P., Goodman, A.W., Pósa, L.: The representation of a graph by set intersections. Canadian Journal of Mathematics **18**, 106–112 (1966)
25. Klamt, S., Haus, U., Theis, F.J.: Hypergraphs and cellular networks. PLoS Computational Biology **5**(5) (2009). doi:10.1371/journal.pcbi.1000385
26. Gallo, G., Longo, G., Pallottino, S.: Directed hypergraphs and applications. Discret. Appl. Math. **42**(2), 177–201 (1993). doi:10.1016/0166-218X(93)90045-P
27. Ausiello, G., Giaccio, R., Italiano, G.F., Nanni, U.: Optimal traversal of directed hypergraphs. ICSI, Berkeley, CA (1992)
28. Gao, J., Zhao, Q., Ren, W., Swami, A., Ramanathan, R., Bar-Noy, A.: Dynamic shortest path algorithms for hypergraphs. IEEE/ACM Trans. Netw. **23**(6), 1805–1817 (2015). doi:10.1109/TNET.2014.2343914
29. Ribeiro, B.F., Basu, P., Towsley, D.: Multiple random walks to uncover short paths in power law networks. In: 2012 Proceedings IEEE INFOCOM Workshops, Orlando, FL, USA, March 25-30, 2012, pp. 250–255 (2012). doi:10.1109/INFCOMW.2012.6193500
30. Głąbowski, M., Musznicki, B., Nowak, P., Zwierzykowski, P.: Shortest path problem solving based on ant colony optimization metaheuristic. Image Processing & Communications **17**(1-2), 7–17 (2012)
31. Li, D.: Shortest paths through a reinforced random walk. Technical report, University of Uppsala (2011)
32. Gallagher, S.R., Goldberg, D.S.: Clustering coefficients in protein interaction hypernetworks. In: ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics. ACM-BCB 2013, Washington, DC, USA, September 22-25, 2013, p. 552 (2013). doi:10.1145/2506583.2506635
33. Mubayi, D., Zhao, Y.: Co-degree density of hypergraphs. J. Comb. Theory, Ser. A **114**(6), 1118–1132 (2007). doi:10.1016/j.jcta.2006.11.006
34. Banerjee, A., Char, A.: On the spectrum of directed uniform and non-uniform hypergraphs. arXiv preprint arXiv:1710.06367 (2017)
35. Yu, W., Sun, N.: Establishment and analysis of the supernetwork model for nanjing metro transportation system. Complexity **2018**, 4860531–1486053111 (2018). doi:10.1155/2018/4860531
36. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440 (1998)
37. 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, pp. 277–291 (2007)
38. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5-8, 2013, Lake Tahoe, Nevada, United States., pp. 3111–3119 (2013). http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality
39. Backstrom, L., Boldi, P., Rosa, M., Ugander, J., Vigna, S.: Four degrees of separation. CoRR **abs/1111.4570** (2011). 1111.4570
40. Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM **38**(11), 39–41 (1995). doi:10.1145/219717.219748
41. Backstrom, L., Boldi, P., Rosa, M., Ugander, J., Vigna, S.: Four degrees of separation. In: Web Science 2012, WebSci '12, Evanston, IL, USA - June 22 - 24, 2012, pp. 33–42 (2012). doi:10.1145/2380718.2380723
42. Milgram, S.: The small world problem. Psychology today **2**(1), 60–67 (1967)
43. Travers, J., Milgram, S.: An experimental study of the small world problem. In: Social Networks, pp. 179–197. Elsevier, Washington, DC (1977)
44. 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, pp. 1–32 (2010). doi:10.1007/978-3-642-23577-1_1