



# Generating Semantic Aspects for Queries

Dhruv Gupta  
Klaus Berberich  
Jannik Strötgen  
Demetrios Zeinalipour-Yazti


MPI-I-2017-5-001 September 2017

## Authors' Addresses

Dhruv Gupta  
Max-Planck-Institut für Informatik  
Saarland Informatics Campus E1 4  
D-66123 Saarbrücken  
Germany   
Saarbrücken Graduate School of Computer Science  
Saarland University  
Saarland Informatics Campus E1 3  
D-66123 Saarbrücken  
Germany

Klaus Berberich  
Max-Planck-Institut für Informatik  
Saarland Informatics Campus E1 4  
D-66123 Saarbrücken  
Germany   
htw saar  
Goebenstraße 40  
66117 Saarbrücken  
Germany

Jannik Strötgen  
Max-Planck-Institut für Informatik  
Saarland Informatics Campus E1 4  
D-66123 Saarbrücken  
Germany

Demetrios Zeinalipour-Yazti  
Max-Planck-Institut für Informatik  
Saarland Informatics Campus E1 4  
D-66123 Saarbrücken  
Germany   
University of Cyprus  
1678 Nicosia  
Cyprus

## **Abstract**

Ambiguous information needs expressed in a limited number of keywords often result in long-winded query sessions and many query reformulations. In this work, we tackle ambiguous queries by providing automatically generated semantic aspects that can guide users to satisfying results regarding their information needs. To generate semantic aspects, we use semantic annotations available in the documents and leverage models representing the semantic relationships between annotations of the same type. The aspects in turn provide us a foundation for representing text in a completely structured manner, thereby allowing for a semantically-motivated organization of search results. We evaluate our approach on a testbed of over 5,000 aspects on Web scale document collections amounting to more than 450 million documents, with temporal, geographic, and named entity annotations as example dimensions. Our experimental results show that our general approach is Web-scale ready and finds relevant aspects for highly ambiguous queries.

## **Keywords**

Temporal Expressions; Named Entities; Geographic Locations; Semantic Annotations; Knowledge Graph; Information Retrieval; Query Intent; Semantic Search; Linking Structured and Unstructured Data

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Challenges . . . . .	3
1.2	Method Outline . . . . .	4
1.3	Applications . . . . .	5
1.4	Contributions and Outline . . . . .	5
<b>2</b>	<b>Background</b>	<b>6</b>
2.1	Semantic Annotations . . . . .	6
2.2	Notation . . . . .	7
2.3	Method . . . . .	9
2.4	Semantic Models . . . . .	9
2.5	Factor Functions . . . . .	11
2.6	Generating Aspects . . . . .	12
<b>3</b>	<b>Evaluation</b>	<b>17</b>
3.1	Setup . . . . .	17
3.2	Results . . . . .	22
<b>4</b>	<b>Related Work</b>	<b>26</b>
<b>5</b>	<b>Conclusions</b>	<b>29</b>

# 1 Introduction

The Web has grown so vast such that precise navigational and transactional queries are now served using named entities in *knowledge graphs* [16] or with the help of curated *knowledge panels* [32]. For instance, for the very specific query `summer olympics rio 2016`, commercial search engines are able to extract and display knowledge panels informing the user about medal tallies and other related information extracted from the Web. However, ambiguous informational queries cannot be served directly. For these vague queries, a session of query reformulations is required to sketch the information need. According to an estimation using a query log, 46% of users reformulate their queries [30]. Thus, there exists a challenge: *guiding users to relevant documents regarding their information needs*. To answer this challenge we introduce the concept of semantic aspects.

Natural language text can be annotated with various kinds of semantic annotations. In particular, there exist named entity recognition and disambiguation (NERD) tools (e.g., AIDA [35]) that can annotate and disambiguate mentions of locations and other named entities to canonical entries in knowledge graphs (e.g., YAGO [45]). Also, temporal expressions in text can be resolved using temporal taggers (e.g., HEIDELTIME [44]) with high precision. These are important semantic annotations in the domain of information retrieval as shown in many studies [27, 47]: 71% of Web queries were found to mention named entities, while 17.1% of Web queries were found to be implicitly temporal in nature. Thus, by looking beyond terms in text, there exists the possibility of deeply understanding the semantics of natural language.

To generate interesting semantic aspects that can guide users in search, we leverage semantic annotations present in documents. As an example search scenario, consider a user trying to research on the Olympic games. An ambiguous query to convey this information need can be: `olympics`. This query may refer to the different Summer Olympic, Winter Olympic or Paralympic games. These potential semantic aspects can thus be conveyed by different semantic annotations present in the pseudo-relevant set of documents: time intervals, e.g., 2008, 2010, 2012, 2014, or 2016; locations, e.g., Beijing, Vancouver, London, Sochi, or Rio de Janeiro; or named entities, e.g., Michael Phelps, Usain Bolt, or Missy Franklin.

## 1.1 Challenges

Semantic aspects for ambiguous queries cannot be generated simply by counting the frequency of these annotations present in pseudo-relevant documents. Interpreting the semantics underlying the annotations is challenging: temporal expressions can be highly uncertain (e.g., `90s`) and two locations or named entities in a knowledge graph can be related by many facts, e.g., ‘Maria Sharapova lives in US but represents Russia in sports [4]. Thus, the aspect generation method must consider the additional complexity of modeling the semantics underlying the annotations. Moreover, queries can signify different kinds of ambiguities: temporal ambiguity (e.g., `tokyo summer olympics` — 1964 or 2020), location ambiguity (e.g., `rome` — many cities in US have towns named after European cities), or entity ambiguity (e.g., `spitz` — Mark or Elisa Spitz). Clearly, once the initial ambiguity associated with a query is identified, other annotations are then useful in the aspect generation process. For instance, for the query `tokyo summer olympics` various named entities and locations can be associated with the two different Olympic games. Thus, the aspect generation method must additionally allow for the flexibility of analyzing the query from many different semantic dimensions. Finally, there currently exists no benchmark for automatic evaluation of generated semantic aspects for highly ambiguous queries. Thus, to close this gap, we provide a novel curated testbed for the research community in the form of a set of informational queries and associated ground truth semantic aspects assimilated from WIKIPEDIA.

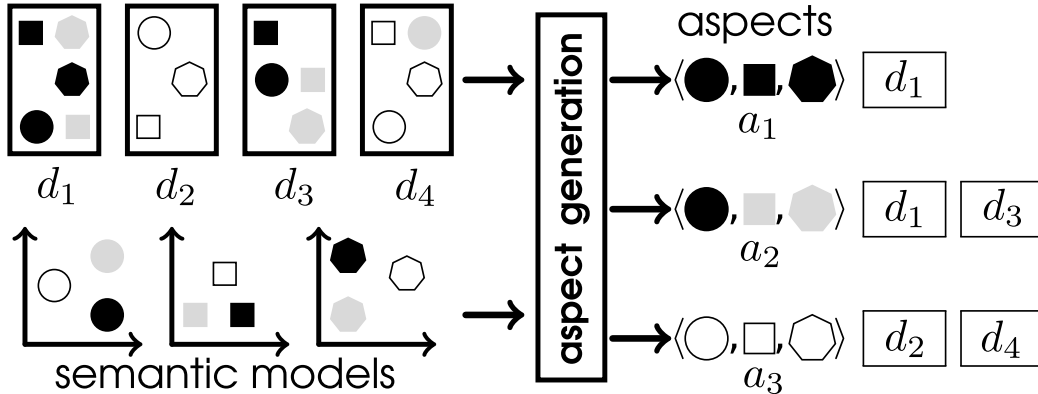


Figure 1.1: Aspect generation process. Given a set of documents with heterogeneous annotations (different colors) of different types (shapes) and semantic models underlying the annotation types, aspects cover combinations of annotations and are associated with relevant documents. The aspects can then be used to explore the documents associated with them, e.g.,  $a_3$  is associated with documents  $d_2$  &  $d_4$ . Conversely, the documents can also be structured using the aspects they generated, e.g.,  $d_1$  can be structured using  $a_1$  &  $a_2$ .

## 1.2 Method Outline

In this work, we consider the problem of automatically generating semantic aspects for resolving ambiguous informational queries. We solve this problem by proposing a novel partitioning algorithm that identifies interesting aspects in the pseudo-relevant set of documents by generating factors in the mathematical models that represent the semantics behind the annotations. The factors are generated by factoring functions that consider the frequency of annotations in their respective semantic models and the relevance of the documents containing them. The partitioning algorithm also allows us to analyze the initial ambiguity behind the query by permuting the order in which the factor functions are applied to the initial set of pseudo-relevant documents. The aspects generated by our algorithm thus represent unstructured text in a completely structured manner. A high-level overview of our proposed method is sketched in Figure 1.1.

<b>Query:</b> olympic medalists
⟨ Time: [2016,2016]
Entities: [YAGO:Ryan_Lochte] [YAGO:Michael_Phelps] [YAGO:Missy_Franklin] [YAGO:Conor_Dwyer] [YAGO:Katie_Ledecky] [YAGO:Aly_Raisman] [YAGO:Simone_Biles] [YAGO:Nathan_Adrian] [YAGO:Alexander_Massialas] [YAGO:Anthony_Ervin] [YAGO:Gabby_Douglas] [YAGO:Sun_Yang]
Locations: [YAGO:United_States] [YAGO:California] [YAGO:New_York_City] [YAGO:Los_Angeles] ⟩

Figure 1.2: An example aspect computed for the query *olympic medalists*. It shows some of the most prominent US swimmers and gymnasts who won numerous gold medals during the 2016 Summer Olympics.

## 1.3 Applications

The aspects lend themselves to various applications in information retrieval. Most importantly, they result in a structured representation of documents, thus facilitating the organization of the documents along various aspects for information consumption. Another direct implication of having a structured representation of documents is search result diversification, whereby we can obtain a subset of search results that cover the identified semantic aspects. The semantic aspects can also act as *knowledge panels* for highly ambiguous queries which current search engines do not serve very well. Exploratory search interfaces can also benefit from the semantic aspects, which can serve as potential search directions for users. An example aspect derived for the query *olympic medalists* by our method is shown in Figure 1.2.

## 1.4 Contributions and Outline

The main contributions of this work and the sections where they are discussed are as follows:

1. In Section 2.3, we describe the algorithm that generates aspects from annotated text while considering the semantic relationships underlying the annotations. The approach is amenable to other domains where annotations other than time, location, and named entities are of importance.
2. In Section 2.3, we also discuss a novel structured representation of documents using the identified aspects, which can be exploited by applications beyond search.
3. In Section 3, we describe a novel testbed of informational queries consisting of over 5,000 aspects derived from WIKIPEDIA used for evaluating our aspect generation process on four large corpora including ClueWeb’09 and ClueWeb’12.



## 2 Background

We begin by describing the different semantic annotations and the notational conventions we use throughout the paper. To make the discussion clear, we use Table 2.1 as our running example.

### 2.1 Semantic Annotations

Our approach for generating semantic aspects from documents is generic. For explanation and evaluation, we consider the following **example dimensions**: temporal expressions, geographic locations, and other named entities (e.g., persons and organizations). Alternative dimensions that could be used to generate other types of semantic aspects in specific domains are: *i.* in the biomedical domain, symptoms, drugs, and side effects in health-related document collections or social media; *ii.* in the e-commerce domain, products, features, and sentiment in texts such as reviews. For our aspect generation process, it is only important that there are annotations of different dimensions and that there is a semantic model underlying each dimension – as will be explained below. First, we briefly describe our example dimensions for generating aspects.

Query: olympic medalists		
Id	Content	Score
$d_1$	hundreds of medals up for grabs for athletes at <u>rio</u> olympics <u>2016</u> .	0.250
$d_2$	<u>michael phelps</u> career spanning <u>'92-'16</u> consists of many medals.	0.250
$d_3$	russian medalist <u>viktor lebedov</u> banned from <u>rio</u> olympics.	0.250
$d_4$	<u>usain bolt</u> has won many medals at <u>2008</u> olympics.	0.125
$d_5$	<u>tokyo</u> will host <u>2020</u> olympics.	0.125

Table 2.1: Sample set of annotated documents.

**Temporal Expressions** in texts can be extracted with temporal taggers. They are able to identify and resolve explicit, implicit, relative, and underspecified temporal expressions given other metadata (e.g., publication dates) [44]. We represent time associated with each aspect as a time interval  $[b, e]$  with begin time point  $b$  and end time point  $e$ . Temporal expressions for documents in Table 2.1 are: 2016, '92-'16, 2008, and 2020.

**Geographic Locations and Other Named Entities** in text are modeled as canonical entries in a knowledge graph such as YAGO [45] or FREEBASE [15]. These annotations are obtained by using *named entity recognition and disambiguation* (NERD) tools, e.g., AIDA [35]. A NERD tool is able to detect the mentions of named entities and further disambiguate them to their canonical entry in a knowledge graph. We differentiate between locations and other named entities, by detecting the presence of a relation in FREEBASE containing its geographic coordinates. Locations in Table 2.1 are `rio` and `tokyo`. While other named entities annotated in Table 2.1 are `michael phelps`, `viktor lebedov`, and `usain bolt`.

## 2.2 Notation

The aspect generation process requires us to retrieve a pseudo-relevant set of documents for a given query. We next describe the framework and notation associated with these steps.

**Retrieving Pseudo-Relevant Documents.** Consider a document collection,

$$\mathcal{D} = \{d_1, d_2, \dots, d_N\},$$

where each document  $d \in \mathcal{D}$  contains a bag of words  $d_{\mathcal{W}}$  drawn from a vocabulary  $\mathcal{V}$ . Each document can be further annotated with different annotations

$$d = \{d_{\mathcal{W}}, d_{\mathcal{X}_1}, d_{\mathcal{X}_2}, \dots, d_{\mathcal{X}_n}\},$$

where  $d_{\mathcal{X}}$  denote the different types of **semantic annotations** or **dimensions** associated with the document. Concretely, the semantic annotations we use are in the form of temporal expressions  $d_{\mathcal{T}}$ , geographic locations  $d_{\mathcal{G}}$ , and other named entities  $d_{\mathcal{E}}$

$$d = \{d_{\mathcal{W}}, d_{\mathcal{T}}, d_{\mathcal{G}}, d_{\mathcal{E}}\}.$$

Thus,  $d_2$  as an example from Table 2.1 can be represented as:  $\{\{\text{michael phelps career span 92 16 consist medal}\}, \{[1992, 2016]\}, \{\emptyset\}, \{\text{Yago:Micheal_Phelps}\}\}$ .

Given a keyword query  $q$ , a set of pseudo-relevant documents  $\mathcal{R}$  for it is retrieved using the method,  $\text{IR}(\bullet)$ . We follow the notational convention in [19] to describe a simple search system:

$$\mathcal{R} = \text{IR}(q, k, \Theta, \mathcal{D}),$$

$\text{IR}(\bullet)$  is a retrieval method wherein the argument  $q$  specifies the query keywords,  $k$  specifies the size of the pseudo-relevant set required,  $\Theta \in \mathbb{R}^m$  specifies a set of parameters relevant for  $\text{IR}(\bullet)$ , and  $\mathcal{D}$  specifies the document collection.  $\text{IR}(\bullet)$  represents a *totally ordered* relation in  $\mathcal{R} \subseteq \mathcal{D}$ . That is, given  $d, d' \in \mathcal{R}$ , it holds that either  $d \preceq d'$  or  $d' \preceq d$  [21], where ties are broken arbitrarily. Internally,  $\text{IR}(\bullet)$  utilizes a  $\text{SCORE}(\bullet)$  function to assign *relevance* of documents to the given query to produce the total order,

$$\text{SCORE}(d_{\mathcal{W}}, \mathcal{R}) : \{d_{\mathcal{W}} \in d \mid \forall d \in \mathcal{R}\} \rightarrow \mathbb{R}^+.$$

A simple  $\text{SCORE}$  function based on word counts and Laplace smoothing can then be designed as follows:

$$\text{SCORE}(d_{\mathcal{W}}, \mathcal{R}) = \prod_{w \in q} \frac{\mathbb{1}(w \in d_{\mathcal{W}}) + 1}{|d_{\mathcal{W}}| + |\mathcal{V}|}.$$

However, other more sophisticated models, e.g., Okapi BM25 could also be used. For the example query `olympic medalists` we obtain  $\mathcal{R} = \{d_1, d_2, d_3, d_4, d_5\}$  as a pseudo-relevant set of documents ordered by their scores.

**Aspect Generation.** Given a set of pseudo-relevant documents  $\mathcal{R}$  for a query  $q$ , an ordered set of *aspects*  $\mathcal{A}$  is to be determined

$$\mathcal{A} = \langle a_1, a_2, \dots, a_n \rangle.$$

An **aspect**  $a \in \mathcal{A}$  consists of **factors**  $a_{x_i}$ , obtained by **factoring functions** that rely on the frequently occurring annotations in their respective semantic models and additionally the co-occurrence of the factors from different dimensions in some subset of  $\mathcal{R}$ ,

$$a = \langle a_{x_1}, a_{x_2}, \dots, a_{x_n} \rangle.$$

Specifically, for our example dimensions, an *aspect* reflects the salience of the *semantic relationship* between a time interval, geographic locations, and named entities by virtue of the frequency in their semantic model as well as the frequency of their co-occurrence in some subset of  $\mathcal{R}$ . Thus, for the example dimensions an aspect  $a \in \mathcal{A}$  is modeled as,

$$a = \langle a_{[b,e]}, a_{\mathcal{G}}, a_{\mathcal{E}} \rangle,$$

where  $a_{[b,e]}$  denotes the time interval,  $a_G$  denotes the set of locations, and  $a_E$  denotes the set of other named entities. One sample aspect that could have been determined for the example shown in Table 2.1 is:  $a_1 = \langle [2016, 2016], \{\text{Yago:Michael\_Phelps}, \text{Yago:Viktor\_Lebedov}\}, \{\text{Yago: Rio\_de\_Janerio}\} \rangle$ .

The aspects which are assimilated from multiple documents can then be used to transform the *semi-structured* documents with annotations into a *structured* representation. This structured representation of documents is then immediately useful for applications in search tasks, such as search result diversification and re-ranking of pseudo-relevant documents.

$$\boxed{\text{semi-structured}} \quad \rightarrow \quad \boxed{\text{structured}}$$

$$d = \{d_W, d_T, d_G, d_E\} \quad \quad \quad d = \langle a_1, a_2, \dots, a_k \rangle$$

## 2.3 Method

In this section, we first describe the mathematical models for the different semantic annotations that we use. These semantic models are utilized for assessing the relationships between annotations of the same type. Thereafter, we describe our aspect generation method — *the partitioning algorithm* — that recursively partitions the pseudo-relevant set of documents by relying on two sets of computation operations. The first computation generates *factors* by counting the frequency of annotations having the same type in their semantic models. The second computation provides us the co-occurrence frequency of factors belonging to the different semantic annotations.

## 2.4 Semantic Models

Our hypothesis is that to find a set of interesting aspects  $\mathcal{A}$  underlying the ambiguous query, we need to consider the frequency of temporal expressions, locations, and other named entities in their respective semantic models, along with their co-occurrence in the documents. A factor in an aspect is deemed interesting by virtue of its frequency. Formally, we define it as follows:

**Definition 1. *Interestingness.*** A factor  $(x \in \mathcal{X})$  is considered interesting if it is frequent in a mathematical model of the semantics for that dimension.

To capture the semantics of annotations is a challenging task. Key design issues that need to be kept in mind regarding semantic models for our example dimensions are:

1. Temporal expressions can indicate an *uncertain* time interval, e.g., 1990s. In such cases, the begin and end of the time interval conveyed is not clear.
2. Temporal expressions can be present at different levels of granularity e.g., day, month, and year granularity.
3. Locations and other named entities may share common relationships, e.g., Tokyo and Beijing both lie in Asia.

## Time Model

Temporal expressions are inherently uncertain. An uncertain temporal expression can thus refer to infinitely many time intervals. The uncertainty in temporal expressions can be modeled by analyzing when the time interval could have begun and ended [11]. That is, the temporal expression 90s can refer to any time interval that can begin ( $b$ ) in [1990, 1999] and end ( $e$ ) in [1990, 1999] (with  $b \leq e$ ). In other words,  $b \in [b_\ell, b_u]$  and  $e \in [e_\ell, e_u]$  giving the *uncertainty-aware time model* [11]:

$$T = \langle b_\ell, b_u, e_\ell, e_u \rangle.$$

We can therefore account for the uncertainty in temporal expressions by representing them in the uncertainty-aware time model. The expression 90s is then represented as:  $\langle 1990, 1999, 1990, 1999 \rangle$ .

## Location and Named Entity Model

Each geographic location and named entity identified by AIDA is linked to its canonicalized entry in the YAGO knowledge graph. Each YAGO entity can further be associated to its originating WIKIPEDIA<sup>1</sup> article. Each WIKIPEDIA article further contains links to other entities' WIKIPEDIA articles, indicating some *semantic relatedness* between the entities or geographic locations. We thus model each canonicalized YAGO geographic location or named entity by its WIKIPEDIA article and the associated link structure to other articles in WIKIPEDIA. Formally, each location or entity  $e$  can be described by the links  $\ell$  its article  $W_e$  shares with other articles in WIKIPEDIA  $W$ ,

$$W_e = \langle \ell_1, \ell_2, \dots, \ell_{|W|} \rangle.$$

---

<sup>1</sup>[www.wikipedia.org](http://www.wikipedia.org)

## 2.5 Factor Functions

We next describe how to find interesting factors associated with each dimension in a set of documents  $\mathcal{R}$  by using its factoring function  $\text{FACTOR}(\mathcal{X}, \mathcal{R})$ .

### Factoring Time - Factor( $\mathcal{X}_{\mathcal{T}}, \mathcal{R}$ )

Temporal expressions in documents can be analyzed to generate interesting time intervals at different levels of granularity. To compute a set of interesting time intervals for a partition of documents, say  $\mathcal{R}$ , we use a method similar to the one described in [28]. Interesting time intervals (factors) can be found by first generating overlaps of the temporal expressions in the uncertainty-aware time model. The time intervals (factors) are then scored by counting the overlaps of temporal expressions in the uncertainty-aware time model weighted by the score of the document containing the temporal expression:

$$\widehat{\text{SCORE}}([b, e], \mathcal{R}) = \sum_{d \in \mathcal{R}} \text{SCORE}([b, e], d_{\mathcal{T}}) \cdot \text{SCORE}(d_{\mathcal{W}}, \mathcal{R}).$$

The function  $\text{SCORE}(d_{\mathcal{W}}, \mathcal{R})$  gives the score of  $d$  with respect to  $q$ . The mapping of the function  $\widehat{\text{SCORE}}([b, e], \mathcal{R})$  can be defined as:

$$\widehat{\text{SCORE}}([b, e], \mathcal{R}) : \{[b, e] \in d_{\mathcal{T}} \mid d \in \mathcal{R}\} \rightarrow \mathbb{R}^+.$$

The function  $\text{SCORE}([b, e], d_{\mathcal{T}})$  then estimates the likelihood of generating the time interval  $[b, e]$  from  $d_{\mathcal{T}}$ . Formally,

$$\text{SCORE}([b, e], d_{\mathcal{T}}) = \frac{1}{|d_{\mathcal{T}}|} \cdot \sum_{T \in d_{\mathcal{T}}} \frac{\mathbf{1}([b, e] \in T)}{|T|}.$$

The cardinality of  $|T|$  indicates the number of time intervals that can be generated from it. The characteristic function  $\mathbf{1}(\bullet)$  then tests the membership of  $[b, e]$  in  $T$ . Therefore, the mapping can be defined as:

$$\text{SCORE}([b, e], d_{\mathcal{T}}) : \{[b, e] \in d_{\mathcal{T}}\} \rightarrow \mathbb{R}^+.$$

### Factoring Locations and Named Entities - Factor( $\mathcal{X}_{\mathcal{G}}, \mathcal{R}$ ) & Factor( $\mathcal{X}_{\mathcal{E}}, \mathcal{R}$ )

To factor locations and other named entities we utilize the concept of *semantic relatedness*. To compute the semantic relatedness we use the *Jaccard overlap* of links shared by the WIKIPEDIA entries of entities  $e$  and  $e'$ , formally

$$\text{EESIM}(e, e') = \frac{|W_e \cap W_{e'}|}{|W_e \cup W_{e'}|}.$$

To identify the interesting sets of entities and locations, we consider the *relatedness* of the entity with other entities weighted by the score of the document that contains them. That is,

$$\widehat{\text{SCORE}}(e, \mathcal{R}) = \sum_{(d \in \mathcal{R})} \text{SCORE}(d_{\mathcal{W}}, \mathcal{R}) \cdot \sum_{(e' \in d_{\mathcal{E}})} \text{EESIM}(e, e').$$

Score of factors containing multiple entities, i.e., having relatedness between them (EESIM) above a given threshold, is equal to the product of the above individual entity scores.

## Normalizing Scores

The range of unnormalized  $\widehat{\text{SCORE}}(\bullet)$  function is  $\mathbb{R}^+$ ; we use the *softmax* function to normalize the scores so that a function’s range is restricted to  $[0, 1]$  [18]:

$$\text{SCORE}(\{x\}, \mathcal{R}) = \frac{e^{\widehat{\text{SCORE}}(\{x\}, \mathcal{R})}}{\sum_{\{x'\} \in \text{FACTOR}(\mathcal{X}, \mathcal{R})} e^{\widehat{\text{SCORE}}(\{x'\}, \mathcal{R})}},$$

where,  $\text{FACTOR}(\mathcal{X}, \mathcal{R})$  denotes the set of factors identified. We choose the softmax function for normalization in order to boost the scores of those factors which have higher non-normalized scores.

## 2.6 Generating Aspects

Our proposed method to generate semantic aspects from a given set of documents is inspired by the *Apriori* algorithm for frequent itemset mining [8]. The Apriori algorithm, however, is not informed of the inherent semantics underlying the annotations and, as such, will not capture any *interesting* relationships amongst them.

Given the pseudo-relevant set of documents  $\mathcal{R}$  for query  $q$  and the three example dimensions,  $\mathcal{X} \in \{\mathcal{X}_t, \mathcal{X}_g, \mathcal{X}_e\}$ , we need to identify *interesting* aspects in the form of  $\langle a_{[b,e]}, a_{\mathcal{G}}, a_{\mathcal{E}} \rangle$ . Enumerating all possible combinations of different annotations in a naïve manner is computationally intractable. To generate these patterns, we propose the following recursive *partitioning algorithm*, that iteratively partitions  $\mathcal{R}$  using factoring functions:

$$\begin{aligned} a_{[b,e]} &= \text{FACTOR}(\mathcal{X}_t, \mathcal{R}), \\ a_{\mathcal{G}} &= \text{FACTOR}(\mathcal{X}_g, \mathcal{R}^{(t)}), \\ a_{\mathcal{E}} &= \text{FACTOR}(\mathcal{X}_e, \mathcal{R}^{(t)(g)}). \end{aligned}$$

Thus, the *partitioning algorithm* induces a new relation:

$$\mathcal{A} \subset 2^{\mathcal{X}_t \times \mathcal{X}_g \times \mathcal{X}_e}.$$

Figure 2.1 illustrates this recursive algorithm and how the aspects are generated from this process. The general linear-recursive definition and corresponding FACTOR methods can be defined as follows.

**Definition 2. Partitioning Algorithm**

Let  $\mathcal{X}_1 \dots \mathcal{X}_n$  be dimensions and  $\mathcal{R}$  a document set. We are asked to find a set of interesting aspects  $\mathcal{A} = \bigcup a = \bigcup \{\{x_1\}, \{x_2\}, \dots, \{x_n\}\}$  that spans across the given dimensions. The set of aspects  $\mathcal{A}$  can be generated by:

$$\begin{aligned} \textbf{Basis Step} : \{x_1\} &= \text{FACTOR}(\mathcal{X}_1, \mathcal{R}) \\ \textbf{Inductive Step} : \{x_k\} &= \text{FACTOR}(\mathcal{X}_k, \mathcal{R}^{(k-1)\dots(1)}) \end{aligned}$$

The *interestingness* of an element in each aspect along a particular dimension is abstractly captured by the FACTOR method by considering a *semantic model* for that dimension. Specifically, for our example dimensions the FACTOR method for time captures the frequency of time intervals in a time model that is informed of *temporal uncertainty* at different levels of granularity. While the FACTOR method for locations and other named entities considers the frequency of their occurrence and also their *relatedness* to the other locations and named entities. Formally, the FACTOR functions identify *interesting* sets of patterns along dimension  $\mathcal{X}$  given a set of documents  $\mathcal{R}$  as:

$$\begin{aligned} \text{FACTOR}(\mathcal{X}_1, \mathcal{R}) : \{d_{\mathcal{X}_1} \in d \mid \forall d \in \mathcal{R}\} &\rightarrow 2^{\mathcal{X}_1}, \\ \text{FACTOR}(\mathcal{X}_k, \mathcal{R}^{(k-1)\dots(1)}) : \{d_{\mathcal{X}_k} \in d \mid \forall d \in \mathcal{R}^{(k-1)\dots(1)}\} &\rightarrow 2^{\mathcal{X}_k}. \end{aligned}$$

Each set of *interesting factors*  $\{x_1\}_i \subseteq 2^{\mathcal{X}_1}$  is associated with a partition  $\mathcal{R}_i^{(1)} \subseteq \mathcal{R}$  that generated  $\{x_1\}_i$ . We create a *partition index* that keeps track of  $\langle \{x_k\}_i, \mathcal{R}_i^{(k-1)\dots(1)} \rangle$ . By concatenating the factors  $\{x_1\}_i, \{x_2\}_i, \dots, \{x_n\}_i$  obtained from the same partition  $\mathcal{R}_i^{(1)(2)\dots(n)}$ , we can identify the *aspects*. The *partitioning algorithm* thus identifies a set of aspects:

$$\mathcal{A} \subset 2^{\mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_n}.$$

The *partitioning algorithm* is still computationally expensive if we were to consider *every* factor along each dimension. To prune the recursion tree, we now define *minimum support* for our algorithm.



### Definition 3. *Minimum Support*

Given a factor function  $\text{FACTOR}(\mathcal{X}, \mathcal{R})$ , let its corresponding scoring function  $\text{SCORE}(\{x\}, \mathcal{R})$  be defined as follows:

$$\text{SCORE}(\{x\}, \mathcal{R}) : \{\{x\} \in d_{\mathcal{X}} \mid \forall d \in \mathcal{R}\} \rightarrow [0, 1].$$

Then, for a given value of minimum support  $\sigma \in [0, 1]$ , a factor is deemed interesting iff:

$$\text{SCORE}(x, \mathcal{R}) \geq \sigma.$$

Thus, the problem of generating the aspects for  $n$  dimensions is defined by  $\text{PARTITION}(\mathcal{R}, \mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n, \sigma)$ . For our example dimensions, the instantiation is  $\text{PARTITION}(\mathcal{R}, \mathcal{X}_t, \mathcal{X}_g, \mathcal{X}_e, \sigma)$ .

## Aspect Scoring

Aspects can further be ranked using the following SCORE function:

$$\text{SCORE}(a, d) = \text{SCORE}(a_{[b,e]}, d) \cdot \text{SCORE}(a_{\mathcal{E}}, d) \cdot \text{SCORE}(a_{\mathcal{G}}, d).$$

With,

$$\begin{aligned} \text{SCORE}(a_{\mathcal{E}}, d) &= \prod_{e \in a_{\mathcal{E}} \cap d_{\mathcal{E}}} \text{SCORE}(e, d), \\ \text{SCORE}(a_{\mathcal{G}}, d) &= \prod_{g \in a_{\mathcal{G}} \cap d_{\mathcal{G}}} \text{SCORE}(g, d). \end{aligned}$$

In order to make the scores of the location and entity dimension comparable with respect to the minimum support, we again normalize them using the softmax function.

## Structured Representation of Documents

For query  $q$  and its corresponding set of pseudo-relevant documents  $\mathcal{R}$ , we now have a set of aspects  $\mathcal{A}$ . As mentioned earlier, each aspect  $a \in \mathcal{A}$  is generated from a partition  $R_i^{(k-1)\dots(1)}$ . From this *partition index*, we can obtain the inverse mapping of documents to aspects. We then have a **structured representation of documents** over the aspects:

$$d = \langle a_1, a_2, \dots, a_k \rangle.$$

For instance, we can now represent documents  $d_2$  and  $d_3$  from Table 2.1 as:  $d_2 = \langle a_1 \rangle$  and  $d_3 = \langle a_1 \rangle$ .

## Query Pivoting

The order in which the factoring functions are applied can result in different sets of interesting aspects. This is due to the fact that *minimum support* in our algorithm is not merely counting *annotations* but is rather realized by the factoring method. Therefore, given three dimensions, we can realize six different sets of interesting aspects by permutation of the different factor methods. This in turn provides us different ways of analyzing the **initial ambiguity** underlying the query, for example:

- ▷ Temporal ambiguity e.g., `tokyo summer olympics`
- ▷ Geographical ambiguity e.g., `rome`
- ▷ Named entity related ambiguity e.g., `spitz`

If the sequence of factor methods is: *time*  $\rightarrow$  *entity*  $\rightarrow$  *geography*, then the resulting set of aspects will be denoted by  $\mathcal{A}_{t \rightarrow e \rightarrow g}$ . The other five possibilities are:  $\mathcal{A}_{t \rightarrow g \rightarrow e}$ ,  $\mathcal{A}_{g \rightarrow t \rightarrow e}$ ,  $\mathcal{A}_{g \rightarrow e \rightarrow t}$ ,  $\mathcal{A}_{e \rightarrow t \rightarrow g}$ , and  $\mathcal{A}_{e \rightarrow g \rightarrow t}$ . Using the illustration in Figure 2.1, these six factor sequences can be obtained by following the different paths in the lattice (the highlighted bold path refers to  $\mathcal{A}_{t \rightarrow g \rightarrow e}$ ).

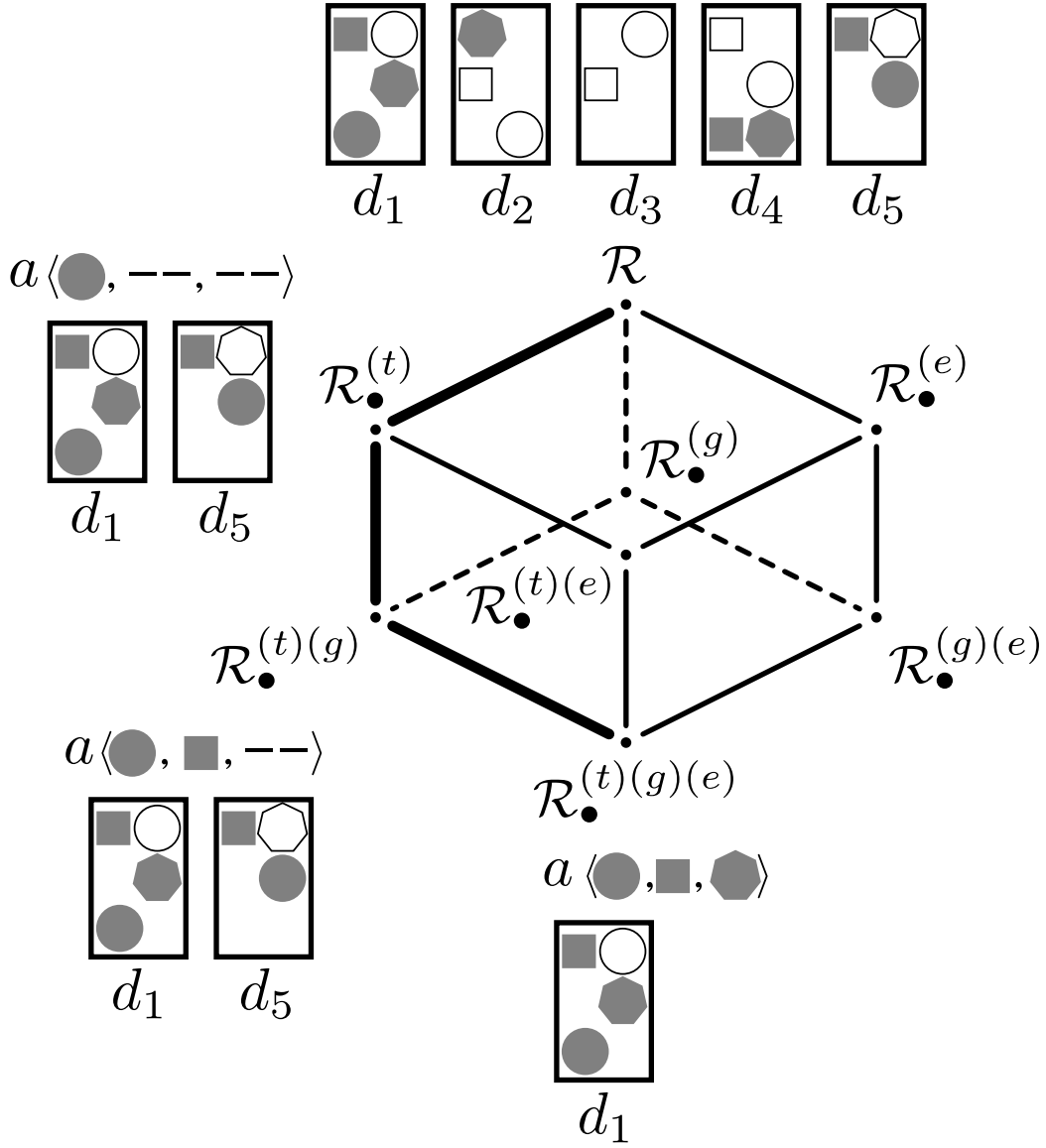


Figure 2.1: The partitioning algorithm. The lattice structure of the aspects generated by our recursive partitioning algorithm is shown. Each element in the lattice corresponds to the partition of documents that arises by applying the factor function for that dimension, e.g.,  $\mathcal{R}_{\bullet}^{(t)}$  is generated by factoring  $\mathcal{R}$  along time. One such time factor  $a = \langle \bullet, \text{---}, \text{---} \rangle$  is generated by documents  $\{d_1, d_5\} \in \mathcal{R}_{\bullet}^{(t)}$ . Continuing in this recursive manner over the geographic dimension  $g$  we get  $a = \langle \bullet, \blacksquare, \text{---} \rangle$ . The sequence of factoring operations can be permuted to obtain different partitions;  $\mathcal{R}_{\bullet}^{(t)(g)(e)}$  corresponds to time  $\rightarrow$  geography  $\rightarrow$  entity (traversing the bold edges of the lattice).

## 3 Evaluation

We next describe the setup for experiments, the results obtained, and their discussion.

### 3.1 Setup

#### Document Collections

We test our algorithm on two different types of document collections. The first category of document collections consists of news articles. News archives have the benefit of being accompanied by rich metadata in the form of accurate publication dates and well-written text. This can aid natural language processing tools to provide more accurate annotations. For example, temporal taggers can resolve *relative* temporal expressions (e.g., **yesterday**) and *implicit* temporal expressions (e.g., **good friday**) with respect to the publication date. We consider two document collections in this category. One of them is a collection of approximately two million news articles published in the New York Times between 1987 and 2007. It is publicly available as the *New York Times Annotated Corpus* [5]. The other one is a collection of approximately four million news articles collected from various online sources during the period of 2013 to 2016 [34], called *Stics*.

The second category of document collections consists of Web pages. Web crawls unlike news articles have unreliable metadata and ill-formed language. This hampers in obtaining high quality semantic annotations for them. For example, we cannot resolve *relative* and *underspecified* temporal expressions, as the *document creation time* for Web pages may not reflect their true publication dates. We consider the two available Web crawls [1, 2] during 2009 and 2012, which are publicly available as ClueWeb’09 and ClueWeb’12 document collections, respectively. Statistics for the document collections are summarized in Table 3.1.

	News Archives		Web Archives	
	New York Times	Stics	ClueWeb'09	ClueWeb'12
Documents	1,679,374	4,075,720	50,220,423	408,878,432
Avg. Time	12.50	10.09	30.59	5.80
Avg. Location	8.65	5.93	9.49	5.61
Avg. Entity	16.25	10.89	8.23	7.74

Table 3.1: Document statistics for the various document collections used in our evaluation. For the example dimensions time, location, and entities we report the average number of annotations found in at most 10,000 documents retrieved for each informational query in our testbed.

## Annotating Documents

Semantic annotations are central to our approach. To obtain them, we utilize publicly available annotations for the document collections or automatically generate them using various tools. For the news archives and for ClueWeb'09, we utilized AIDA [35], which performs named entity recognition and disambiguation. Each disambiguated named entity is linked to its canonical entry in the YAGO knowledge graph. As a subset of these named entities, we can obtain *geographic locations*. For ClueWeb'12, we utilized the FACC annotations [24] provided by GOOGLE. The FACC annotations contain the offsets of high precision entities spotted in the web pages. Temporal expressions for all the document collections were obtained using the HEIDELTIME temporal tagger [44]. In Table 3.1, we additionally report the average counts of the three types of semantic annotations found in at most 10,000 documents retrieved for each query in our testbed.

## Collecting Ground Truth Aspects for Queries

To evaluate our system, we extracted over 5,000 aspects from WIKIPEDIA. This was done considering their diversity along three dimensions of time, locations, and other named entities for a set of twenty-five *informational queries*. The broad topics of the aspects along with the specific keyword queries and the number of aspects generated are listed in Table 3.2.

<b>(<math>\mathcal{A}_{e \rightarrow t \rightarrow g}</math> &amp; <math>\mathcal{A}_{e \rightarrow g \rightarrow t}</math>)</b>   <b>Achievements</b> <b>[1,508]:</b> nobel prize [114]   olympic medalists [48]   oscars [1,167]   paralympic medalists [24]   space shuttle missions [155]
<b>(<math>\mathcal{A}_{g \rightarrow e \rightarrow t}</math> &amp; <math>\mathcal{A}_{g \rightarrow t \rightarrow e}</math>)</b>   <b>Disasters</b> <b>[1,536]:</b> aircraft accidents [513]   avalanches [56]   earthquakes [39]   epidemics [211]   famines [133]   genocides [35]   hailstorms [39]   landslides [85]   nuclear accidents [26]   oil spills [140]   tsunamis [88]   volcanic eruptions [171]
<b>(<math>\mathcal{A}_{t \rightarrow e \rightarrow g}</math> &amp; <math>\mathcal{A}_{t \rightarrow g \rightarrow e}</math>)</b>   <b>Politics</b> <b>[2,078]:</b> assassinations [130]   cold war [81]   corporate scandals [44]   proxy wars [34]   united states presidential elections [57]   terror attacks [316]   treaties [1,057]   wars [359]

Table 3.2: Categories of query keywords with aspect counts (in brackets) and appropriate sequence of factor operations.

For each query, we constructed a set of ground-truth aspects by considering the table of events present on the WIKIPEDIA page corresponding to the query [3, 12]. For the table, we considered each row consisting of time, locations, and other entities as an aspect. If no locations or named entities were mentioned, we extracted them from the associated *event* page of the row, by running AIDA on the introductory paragraph of the event’s WIKIPEDIA page. For instance, consider the Table 3.3 as an example table of events present on a hypothetical Wikipedia page for Olympic medalists. Treating each row as a ground truth aspect, we look for temporal expressions, e.g., [2008, 2016] as a time factor; locations, e.g., Beijing, London, and Rio de Janeiro as a location factor; and other named entities, e.g., Usain Bolt as entity factor. Similarly, for the second row in Table 3.3, the extracted aspect is:  $\langle [2004, 2016], \{Yago: Athens, Yago: Beijing, Yago: London, Yago: - Rio\_de\_Janeiro\}, \{Yago: Michael\_Phelps\} \rangle$ . This testbed is made available to the research community at the following URL:

<http://resources.mpi-inf.mpg.de/dhgupta/data/aspects2017/>

Years Active	Description	Locations
2008 - 2016	Usain Bolt won total of 9 Olympic medals during the Summer Olympic games in the years he was active.	Beijing, London, and Rio de Janeiro
2004 - 2016	Michael Phelps has won a record number of 23 gold medals at various Olympic games during his career.	Athens, Beijing, London, and Rio de Janeiro

Table 3.3: An example table of events for explaining the assimilation of ground truth for the query: *olympic medalists*.

## Measures

Given a query and its set of pseudo-relevant documents, our algorithm outputs a set of interesting aspects. Two key *characteristics* for evaluation are then to see if the aspects are *correct* with respect to a ground truth and how *novel* the aspects are with respect to other aspects. These two characteristics taken together ensure that our sets of aspects are *meaningful* and *non-redundant*. We next describe the respective two measures *correctness* and *novelty*.

*Similarity* computation between aspects is central to both the correctness and novelty measure. To compute the similarity between the two aspects,  $a$  and  $b$ , we consider their *similarity* dimension-wise. More specifically,

$$\text{SIMILARITY}(a, b) = \frac{1}{3} \left( \frac{|a_{[b,e]} \cap b_{[b,e]}|}{|a_{[b,e]}|} + \frac{|a_{\mathcal{E}} \cap b_{\mathcal{E}}|}{|a_{\mathcal{E}}|} + \frac{|a_{\mathcal{G}} \cap b_{\mathcal{G}}|}{|a_{\mathcal{G}}|} \right),$$

where, for temporal similarity we coarsen the time intervals at year granularity to make them comparable. The temporal overlaps are computed using the uncertainty-aware time model [11] by converting the time intervals to the four-tuple notation. While for the other two dimensions the similarity is akin to computing the *Jaccard similarity* between bag-of-locations and bag-of-entities, however, only with respect to the ground truth (in the denominator).

*Correctness*. Given a set of aspects  $\mathcal{A}$  generated by our algorithm for a query  $q$  and the set of aspects  $\mathcal{B}$  corresponding to the ground truth derived

from WIKIPEDIA page for the same query, correctness can be formalized as:

$$\text{CORRECTNESS}(\mathcal{A}, \mathcal{B}) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \text{SIMILARITY}(a, b).$$

*Novelty* for the set of aspects  $\mathcal{A}$  can be intuitively thought of measuring the *dissimilarity* with respect to  $\mathcal{A}$  itself:

$$\text{NOVELTY}(\mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \frac{1}{|\mathcal{A}|} \sum_{(a' \in \mathcal{A}/\{a\})} \left(1 - \text{SIMILARITY}(a, a')\right).$$

A probabilistic interpretation of the above measures can be arrived at by considering a random draw from the set  $a \in \mathcal{A}$  and another random draw from the set  $b \in \mathcal{B}$  and computing the similarity between  $a$  and  $b$ , thus giving us the likelihood of two aspects being similar. We can additionally conform the *correctness* measure to the standard information retrieval measures such as *precision* and *recall* as follows:

$$\text{PRECISION} = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \max_{b \in \mathcal{B}} \left( \text{SIMILARITY}(a, b) \right),$$

and

$$\text{RECALL} = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \max_{a \in \mathcal{A}} \left( \text{SIMILARITY}(a, b) \right).$$

## Baselines and Systems

We consider two baselines to compare our proposed approach. As a naïve baseline, we treated each document in the pseudo-relevant set to represent an aspect by considering the earliest and latest time point in the document as the aspect’s time interval and bag-of-locations and bag-of-entities to represent the other two dimensions. As a second baseline, we consider *latent Dirichlet allocation* (LDA) [14] to discover  $k$  topics from the pseudo-relevant set of documents. For each topic discovered by LDA, we then consider the top- $(k - 1)$  most relevant documents associated with each topic. From this *partition* of top- $(k - 1)$  documents, we derive the corresponding aspect by considering the earliest and latest time point in the partition as the aspect’s time interval and bag-of-entities and bag-of-locations to represent the two remaining dimensions. We refer to each instantiation of this baseline as LDA- $k$ . For our algorithm, denoted by PA, we considered the specific sequence of factor operations that were deemed meaningful for that query (as shown in Table 3.2) for aspect generation. For instance, since the query **earthquakes** is oriented towards geographic locations we considered the factor sequence operations  $\mathcal{A}_{g \rightarrow e \rightarrow t}$  and  $\mathcal{A}_{g \rightarrow t \rightarrow e}$  for aspect generation.



## Parameters

For each query in Table 3.2, we retrieve at most 10,000 documents with *disjunctive* operator using Okapi BM25 as the retrieval method. We used the standard parameters,  $b = 0.75$  and  $k_1 = 1.20$ , for its configuration. For the LDA baseline, we followed Griffiths and Steyvers [26] for setting its parameters. Specifically,  $\beta$  was set to 0.1 and  $\alpha$  was set to  $50/|topics|$ . We considered three topic set sizes for LDA namely,  $|topics| \in \{50, 100, 200\}$  and the same number of top- $k$  documents for each topic, e.g., for  $|topics| = 50$ , we picked top-50 documents for each topic as its generating partition. For our proposed algorithm, we consider the following global parameters: entity-entity relatedness greater than 0.05 to identify meaningful geographic and other named entity factors; the top 90<sup>th</sup> percentile of the time intervals of interest as time factors; and the minimum support was set to  $\sigma = 0.001$ . The parameters were derived at by observing their effect on few sample queries such as wars and nuclear accidents only on the news archive collections.

## 3.2 Results

We report the results of the experiments over the four different document collections, including two Web collections. In addition to the measures discussed, we report the average number of aspects discovered by our algorithm (PA) and the baselines (BM25 & LDA).

### Results for News Archives

We first consider the results of the systems in terms of correctness and novelty as reported in Table 3.4. For the New York Times collection, our method identifies the most correct aspects with respect to the ground truth as compared to the baselines across all possible factor sequence operations. Despite the observation that Okapi BM25 wins in terms of novelty by considering all pseudo-relevant documents, our method still achieves a high degree of novelty, thereby identifying the most non-redundant set of aspects and is able to partition the set of pseudo-relevant documents to the greatest degree. For the Stics news collection, our method outperforms the baselines in terms of correctness. Okapi BM25 achieves a higher novelty value, however, the increase compared to our method is not significant. Observing both correctness and novelty our method excels in providing both relevant and non-redundant sets of aspects when comparing to the baselines which can only achieve high novelty.

	New York Times			Stics		
	Avg. $ \mathcal{A} $	Correctness	Novelty	Avg. $ \mathcal{A} $	Correctness	Novelty
<b>BM25</b>	3, 3379	0.008	<b>0.442</b>	3,796	0.008	<b>0.434</b>
<b>LDA-50</b>	50	0.005	0.279	50	0.010	0.290
<b>LDA-100</b>	100	0.004	0.228	100	0.008	0.240
<b>LDA-200</b>	200	0.004	0.163	200	0.006	0.163
<b>PA</b>	1, 638	<b>0.013</b>	0.404	480	<b>0.018</b>	0.395

Table 3.4: System results when measuring correctness and novelty on news archives. Best performing systems for the different measures are highlighted in bold.

Now consider precision and recall for the systems, as reported in Table 3.5. For the New York Times, while considering precision, our system consists of more relevant aspects compared to the baselines. With respect to recall, it is at par with the Okapi BM25 baseline. Taking both precision and recall together, our system presents a balanced performance: high precision and recall while the baselines achieve higher recall only. For the Stics corpus, when considering precision and recall, our method again has significant improvements over the baselines. Thus, by taking all the four measures, correctness, novelty, precision, and recall, our method allows us to distill interesting aspects which can guide the user to navigate through a large number of documents.

	New York Times			Stics		
	Avg. $ \mathcal{A} $	Precision	Recall	Avg. $ \mathcal{A} $	Precision	Recall
<b>BM25</b>	3,379	0.098	<b>0.152</b>	3,796	0.072	0.113
<b>LDA-50</b>	50	0.041	0.015	50	0.063	0.031
<b>LDA-100</b>	100	0.034	0.010	100	0.048	0.021
<b>LDA-200</b>	200	0.031	0.007	200	0.046	0.013
<b>PA</b>	1,638	<b>0.264</b>	0.148	480	<b>0.229</b>	<b>0.134</b>

Table 3.5: System results when measuring precision and recall on news archives. Best performing systems for the different measures are highlighted in bold.

## Results for Web Collections

Web collections give us more challenging documents to test the effectiveness of our approach. Particularly, since they are not well-formed, they have a lower average number of annotations per document, the annotations in them are prone to more errors, and the size of the Web collections is magnitudes larger than news archives. Hence, they present a challenging real world scenario to test our methods. We first consider the results for the Web collections when measuring correctness and novelty that are reported in Table 3.6. For ClueWeb’09, our method outperforms both baselines in terms of correctness and novelty. In particular, for novelty our method outperforms the baselines significantly. For ClueWeb’12 our method performs at par with baselines in terms of correctness and novelty. When considering the measures in isolation, for correctness the LDA baseline wins over our method and Okapi BM25 baseline has higher novelty than our method. However, when considering both correctness and novelty together, our system is consistent in providing more correct and novel aspects as opposed to the baselines.

	ClueWeb’09			ClueWeb’12		
	Avg. $ \mathcal{A} $	Correctness	Novelty	Avg. $ \mathcal{A} $	Correctness	Novelty
<b>BM25</b>	9,579	0.009	0.398	9,752	0.012	<b>0.461</b>
<b>LDA-50</b>	50	0.012	0.331	50	<b>0.018</b>	0.350
<b>LDA-100</b>	100	0.009	0.289	100	0.012	0.306
<b>LDA-200</b>	200	0.006	0.246	200	0.010	0.257
<b>PA</b>	1,480	<b>0.014</b>	<b>0.431</b>	529	0.016	0.419

Table 3.6: System results when measuring correctness and novelty on Web collections. Best performing systems for the different measures are highlighted in bold.

Next, we consider the second set of experimental results for Web collections when measuring precision and recall that are reported in Table 3.7. For ClueWeb’09, our method in terms of precision and recall outperforms both baselines significantly. For ClueWeb’12, our method outperforms both baselines with respect to precision. However, in terms of recall Okapi BM25 outperforms our method when considering all the pseudo-relevant documents. Despite of this, our method provides a balanced performance with high precision and moderate recall as compared to the baselines which have high recall but very low precision.

	ClueWeb'09			ClueWeb'12		
	Avg. $ \mathcal{A} $	Precision	Recall	Avg. $ \mathcal{A} $	Precision	Recall
<b>BM25</b>	9,579	0.080	0.152	9,752	0.081	<b>0.256</b>
<b>LDA-50</b>	50	0.073	0.087	50	0.096	0.106
<b>LDA-100</b>	100	0.056	0.075	100	0.071	0.074
<b>LDA-200</b>	200	0.043	0.055	200	0.058	0.049
<b>PA</b>	1,480	<b>0.137</b>	<b>0.178</b>	529	<b>0.234</b>	0.156

Table 3.7: System results when measuring precision and recall on Web collections. Best performing systems for the different measures are highlighted in bold.

## Query Pivoting

The initial factoring dimension addresses the different types of dimensional ambiguity in queries during the aspect generation process. We had partitioned the testbed of queries, according to most meaningful initial factoring dimension. We next describe few queries that had high *correctness* values for the three different types of initial factoring operations to support our hypothesis. For these examples, we specifically investigate the results for the news archives as they had a higher average number of annotations per document. For *time*, the queries that obtained high correctness scores were **nuclear accidents**, **corporate scandals**, and **wars**. The query **wars** in particular for the New York Times also had higher correctness values for *entity* as the initial factoring dimension. For *geographic* dimension, the queries that benefit by factoring this dimension were **epidemics**, **olympic medalists**, and **oil spills**. For the *entity* dimension, the queries **nobel prize**, **nuclear accidents**, and **cold war** achieved high correctness scores.

## Summary

Our experiments on two large news archives show that semantic annotations in the form of temporal expressions, locations, and other named entities can be used to identify semantic aspects that are *correct* and *novel* for document exploration. On Web-scale corpora, where quality annotations are few, our methods can also identify *precise* aspects for information consumption. In addition to this, our method can resolve the different types of initial ambiguity behind *informational queries* by leveraging the semantic models behind the different dimensions.

## 4 Related Work

In this section, we discuss related works.

*Structuring Text for Search.* One of the seminal works in inducing a structure to text was suggested by Hearst and Plaunt [31]: `TEXTTILING`, an algorithm for identifying coherent passages (subtopics) in text documents. By leveraging more recent advances in natural language processing, Koutrika et al. looked at a similar problem of generating *reading orders* [37]. Given, a set of documents, the authors utilized LDA to identify topics in documents for their structured representation. With this representation, documents are hierarchically arranged in a tree, based on their topical generalization and overlap. A path in the generated tree then gives the user a reading order over the documents. However, both approaches were not informed of semantic annotations, which we exploit to identify *aspects* for structuring text for search.

*Faceted Search* systems allow a user to navigate document collections and prune irrelevant documents by displaying important features about them. Going beyond this basic model, Ben-Ytzhak et al. discussed various algorithms that allowed for business intelligence aggregations and more advanced dynamic discovery of correlated facets across multiple dimensions [10]. In contrast to their work, our approach considers semantic models for each dimension during the aspect identification process to cover relations between annotations of the same dimension. Li et al. [38] leveraged semantic metadata present in `WIKIPEDIA` such as entities and their associated category for automated generation of facets for exploring `WIKIPEDIA` articles. While they leveraged only collection-specific knowledge to generate the *facets*, our approach is amenable to multiple dimensions for *aspect* generation. Specifically, for applications in information retrieval, Dou et al. [23] and Kong and Allan [36] investigated how to mine keyword lists present in pseudo-relevant documents for aspect generation. However, in their work semantic annotations were not considered part of their methods. Arenas et al. [25] considered the use of named entities and their relationships in graphs for

generating facets in DBPEDIA abstracts. Their approach however does not incorporate other annotations such as temporal expressions, which we have considered in our work.

*Online Analytical Processing* (OLAP) relies on using *concept hierarchies* associated with data *attributes* to allow for various kinds of analytical aggregations (e.g., drill-up and drill-down). Thus, by going one step further than *faceted search*, OLAP has the potential to retrieve documents that satisfy constraints over an aggregation of dimensions [22]. Zhang et al. further structured text in an OLAP data cube by inducing and associating a *topic hierarchy* using *probabilistic latent semantic analysis* (pLSA) [46]. This allows for OLAP operations over text using the pLSA topic hierarchy. All the OLAP methods can be considered static, i.e., they assume that the document collection is pre-structured with respect to the annotations and their corresponding concept hierarchy. To identify interesting insights the decision maker is required to formulate precise queries. Our work, however, generates aspects that can be used to navigate documents in a dynamic fashion. Since they are generated automatically, the user, need not specify any specific operation.

*Semantic Search.* Rich metadata in the form of disambiguated named entities has enabled retrieval systems to tap into power of curated knowledge graphs such as FREEBASE [15] and YAGO [45]. Bast and Buchhold [9] addressed the problem of jointly indexing text and the contextual knowledge graph entities with their relations. Hoffart et al. [33, 34] demonstrated a semantic search system that offers the capability to retrieve documents and perform analytics via queries composed of keywords, named entities, and their semantic types. All these approaches, however, do not aim at mining insightful aspects for document exploration.

*Event Search.* Time, locations, and entities naturally lend themselves for meaningful representation of events. Spitz and Gertz [43] discussed an approach for graph construction using time points, locations, and named entities for cross-document event summarization. However, their approach analyzed *textual proximity* between annotations and was not informed of the inherent semantics conveyed by the annotations. On the other hand, the method proposed by Gupta et al. [29] also considers semantic annotations, but is not amenable to Web scale. Also, in their approach there is no provision to analyze the dependence between annotations with respect to initial ambiguity of the query. For example, an entity-oriented query (e.g., `clinton`) and a time-oriented query (e.g., `olympics`) cannot be differentiated with respect to the initial ambiguity.

*Exploratory Search.* Exploratory Search systems such as proposed in [17] and [40] address interface design issues that users face when navigating large document collections. Ruotsalo et al. describe a visualization consisting of a radar on which keywords can be followed to change the focus of search [40]. Bozzon et al. describe a query-by-example interface where search directions are manually input for various dimensions pertinent to the domain [17]. In contrast to these approaches, our work automatically identifies search directions in the form of semantic aspects.

*Search Result Diversification.* Diversifying search results is an established way of alleviating the problems associated with ambiguous queries. By diversifying search results, the user is presented with a novel and non-redundant subset of documents distilled from the initial set of pseudo-relevant documents. Existing diversification approaches such as maximum marginal relevance (MMR) [19] and PM-2 [20] utilize implicit text to identify a diverse subset of documents. On the other hand, IA-Select [7] and xQuAD [41] rely on an explicitly specified taxonomy of categories or aspects mined from query suggestions to diversify search results. For implicit diversification, resolving ambiguity is limited to utilizing textual cues. On the other hand, for explicit diversification, alleviating ambiguity relies on document aspects (e.g., query suggestions) to be provided from an external resource. Our generated semantic aspects thus serve as novel topics that are useful for diversification of documents.

*Entity Linking and Entity-Oriented Search.* Linking named entities in queries to the document text in which they occur and subsequently leveraging the context and co-occurring entities has received ample attention in the information retrieval community. Reinanda et al. [39] proposed how to leverage search engine query logs to mine and suggest entities of relevance given entity-oriented query. The authors consider metadata associated with the queries in the query log for their approach e.g., user clicks, user sessions, and query issue timestamps. Their proposed method however does not tap into the document contents for aspect generation. Blanco et al. [13] propose how to connect named entities in queries with their mentions in the documents in which they occur in an efficient manner. Similarly, Schuhmacher et al. [42] propose a method for recommending related entities for entity-centric queries using pseudo-relevance feedback from retrieved documents and knowledge graphs. Our work in contrast, is not limited to named entities and we additionally consider temporal expressions in the document contents. As discussed, temporal expressions in document contents pose several challenges such as being uncertain and being present at several granularities. Prior approaches however do not take into account other dimensions for aspect discovery e.g., temporal expressions.

## 5 Conclusions

In this work, we discussed a novel *partitioning algorithm* that leverages semantic annotations such as temporal expressions, geographic locations, and other named entities to generate *semantic aspects*. The algorithm consists of factor functions which realize the mathematical models behind the semantic annotations to compute their *interestingness*. Further, the factor functions can be applied in different orders to identify the most relevant set of aspects and to disambiguate the different types of ambiguities underlying the query. The set of aspects identified can then lend themselves to a structured representation of documents. Hence, building a foundation for further *retrieval techniques* such as search result diversification to be carried out. Our framework is generic and can accommodate different types of factor functions for other domains such as e-commerce or the medical domain. Our experiments on different types of document collections, including ClueWeb'09 and ClueWeb'12, show that our approach to the problem allows the user to navigate through messy unstructured data in a structured manner at Web scale.



# Bibliography

- [1] The Clueweb09 dataset.  
<http://lemurproject.org/clueweb09/>  
[Online; accessed 13-September-2017].
- [2] The Clueweb12 dataset.  
<http://lemurproject.org/clueweb12/>  
[Online; accessed 13-September-2017].
- [3] List of lists of lists.  
[https://en.wikipedia.org/wiki/List\\_of\\_lists\\_of\\_lists](https://en.wikipedia.org/wiki/List_of_lists_of_lists)  
[Online; accessed 13-September-2017].
- [4] Maria Sharapova.  
[https://en.wikipedia.org/wiki/Maria\\_Sharapova](https://en.wikipedia.org/wiki/Maria_Sharapova)  
[Online; accessed 13-September-2017].
- [5] The New York Times Annotated corpus.  
<https://catalog ldc.upenn.edu/LDC2008T19>  
[Online; accessed 13-September-2017].
- [6] Wikipedia - the free Encyclopedia.  
<https://www.wikipedia.org/>  
[Online; accessed 13-September-2017].
- [7] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*, pages 5–14, 2009.
- [8] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of 20th International Conference on Very Large Data Bases, VLDB 1994, Santiago de Chile, Chile, September 12-15, 1994*, pages 487–499, 1994.

- [9] H. Bast and B. Buchhold. An index for efficient semantic full-text search. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM 2013, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 369–378, 2013.
- [10] O. Ben-Yitzhak, N. Golbandi, N. Har’El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. J. Shekita, B. Sznajder, and S. Yegorov. Beyond basic faceted search. In *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*, pages 33–44, 2008.
- [11] K. Berberich, S. J. Bedathur, O. Alonso, and G. Weikum. A language modeling approach for temporal information needs. In *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010*, pages 13–25, 2010.
- [12] C. S. Bhagavatula, T. Noraset, and D. Downey. *TabEL: Entity Linking in Web Tables*, pages 425–441. Springer International Publishing, Cham, 2015.
- [13] R. Blanco, G. Ottaviano, and E. Meij. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, China, February 2-6, 2015*, pages 179–188, 2015.
- [14] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
- [15] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250, 2008.
- [16] I. Bordino, M. Lalmas, Y. Mejova, and O. V. Laere. Beyond entities: promoting explorative search with bundles. *Inf. Retr. Journal*, 19(5):447–486, 2016.
- [17] A. Bozzon, M. Brambilla, S. Ceri, and P. Fraternali. Liquid query: Multi-domain exploratory search on the web. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, NC, USA, April 26-30, 2010*, pages 161–170, 2010.

- [18] J. S. Bridle. *Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition*, pages 227–236. Springer Berlin Heidelberg, Berlin, Heidelberg, 1990.
- [19] J. G. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1998, Melbourne, Australia, August 24-28 1998*, pages 335–336, 1998.
- [20] V. Dang and W. B. Croft. Term level search result diversification. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2013, Dublin, Ireland, July 28 - August 01, 2013*, pages 603–612, 2013.
- [21] B. A. Davey and H. A. Priestley. *Introduction to lattices and order*. Cambridge University Press, Cambridge, 1990.
- [22] B. Ding, B. Zhao, C. X. Lin, J. Han, C. Zhai, A. N. Srivastava, and N. C. Oza. Efficient keyword-based search for top-k cells in text cube. *IEEE Trans. Knowl. Data Eng.*, 23(12):1795–1810, 2011.
- [23] Z. Dou, S. Hu, Y. Luo, R. Song, and J. Wen. Finding dimensions for queries. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 1311–1320, 2011.
- [24] M. R. Evgeniy Gabilovich and A. Subramanya. Facc1: Freebase annotation of cluweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0), June 2013. <http://lemurproject.org/cluweb12/>.
- [25] B. C. Grau, E. Kharlamov, S. Marciuska, D. Zheleznyakov, and M. Arenas. Semfacet: Faceted search over ontology enhanced knowledge graphs. In *Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference, ISWC 2016, Kobe, Japan, October 19, 2016*, 2016.
- [26] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.
- [27] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 267–274, 2009.

- [28] D. Gupta and K. Berberich. Identifying time intervals of interest to queries. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1835–1838, 2014.
- [29] D. Gupta, J. Strötgen, and K. Berberich. Eventminer: Mining events from annotated documents. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12-16, 2016*, pages 261–270, 2016.
- [30] M. A. Hearst. *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.
- [31] M. A. Hearst and C. Plaunt. Subtopic structuring for full-length document access. In *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1993, Pittsburgh, PA, USA, June 27 - July 1, 1993*, pages 59–68, 1993.
- [32] J. Henry. Providing knowledge panels with search results, May 2 2013. US Patent App. 13/566,489.
- [33] J. Hoffart, D. Milchevski, and G. Weikum. AESTHETICS: analytics with strings, things, and cats. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 2018–2020, 2014.
- [34] J. Hoffart, D. Milchevski, and G. Weikum. STICS: searching with strings, things, and cats. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2014, Gold Coast, QLD, Australia, July 06 - 11, 2014*, pages 1247–1248, 2014.
- [35] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenaу, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, Edinburgh, UK, 27-31 July 2011*, pages 782–792, 2011.
- [36] W. Kong and J. Allan. Extracting query facets from search results. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2013, Dublin, Ireland, July 28 - August 01, 2013*, pages 93–102, 2013.

- [37] G. Koutrika, L. Liu, and S. J. Simske. Generating reading orders over document collections. In *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, pages 507–518, 2015.
- [38] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das. Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, NC, USA, April 26-30, 2010*, pages 651–660, 2010.
- [39] R. Reinanda, E. Meij, and M. de Rijke. Mining, ranking and recommending entity aspects. In *Proceedings of the 38th International ACM Conference on Research and Development in Information Retrieval, SIGIR 2015, Santiago, Chile, August 9-13, 2015*, pages 263–272, 2015.
- [40] T. Ruotsalo, G. Jacucci, P. Myllymäki, and S. Kaski. Interactive intent modeling: Information discovery beyond search. *Commun. ACM*, 58(1):86–92, Dec. 2014.
- [41] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, NC, USA, April 26-30, 2010*, pages 881–890, 2010.
- [42] M. Schuhmacher, L. Dietz, and S. P. Ponzetto. Ranking entities for web queries through text and knowledge. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1461–1470, 2015.
- [43] A. Spitz and M. Gertz. Terms over LOAD: leveraging named entities for cross-document extraction and summarization of events. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 503–512, 2016.
- [44] J. Strötgen and M. Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013.
- [45] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semant.*, 6(3):203–217, Sept. 2008.

- [46] D. Zhang, C. Zhai, and J. Han. Topic cube: Topic modeling for OLAP on multidimensional text databases. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, Sparks, NV, USA, April 30 - May 2, 2009*, pages 1124–1135, 2009.
- [47] R. Zhang, Y. Konda, A. Dong, P. Kolari, Y. Chang, and Z. Zheng. Learning recurrent event queries for web search. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, Massachusetts, USA, October 9-11, 2010*, pages 1129–1139, 2010.

Below you find a list of the most recent research reports of the Max Planck Institut for Informatics. Most of them are accessible via WWW using the URL <http://www.mpi-inf.mpg.de/publications/research-reports/>. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max Planck Institut für Informatik  
– Library and Publications –  
Campus E 1 4  
66123 Saarbrücken  
Germany

E-mail: [library@mpi-inf.mpg.de](mailto:library@mpi-inf.mpg.de)

---

MPI-I-2016-5-002	A. Mishra, K. Berberich	Leveraging Semantic Annotations to Link Wikipedia and News Archives
MPI-I-2016-5-001	D. Gupta, K. Berberich	Diversifying Search Results Using Time
MPI-I-2016-4-002	S. Sridhar, G. Bailly, E. Heydrich, A. Oulasvirta, C. Theobalt	FullHand: Markerless Skeleton-based Tracking for Free-Hand Interaction
MPI-I-2016-4-001	S. Sridhar, F. Müller, M. Zollhöfer, D. Casas, A. Oulasvirta, C. Theobalt	Real-time Joint Tracking of a Hand Manipulating an Object from RGB-D Input
MPI-I-2014-RG1-002	P. Baumgartner, U. Waldmann	Hierarchic Superposition with Weak Abstraction
MPI-I-2014-5-002	A. Anand, I. Mele, S. Bedathur, K. Berberich	Phrase Query Optimization on Inverted Indexes
MPI-I-2014-5-001	M. Dylla, M. Theobald	Learning Tuple Probabilities in Probabilistic Databases
MPI-I-2014-4-002	S. Sridhar, A. Oulasvirta, C. Theobalt	Fast Tracking of Hand and Finger Articulations Using a Single Depth Camera
MPI-I-2014-4-001	K. Kim, J. Tompkin, C. Theobalt	Local High-order Regularization on Data Manifolds
MPI-I-2013-5-002	F. Makari, B. Awerbuch, R. Gemulla, R. Khandekar, J. Mestre, M. Sozio	A Distributed Algorithm for Large-scale Generalized Matching
MPI-I-2013-1-001	C. Huang, S. Ott	New Results for Non-preemptive Speed Scaling
MPI-I-2012-RG1-002	A. Fietzke, E. Kruglov, C. Weidenbach	Automatic Generation of Invariants for Circular Derivations in SUP(LA) 1
MPI-I-2012-RG1-001	M. Suda, C. Weidenbach	Labelled Superposition for PLTL
MPI-I-2012-5-004	F. Alvanaki, S. Michel, A. Stupar	Building and Maintaining Halls of Fame Over a Database
MPI-I-2012-5-003	K. Berberich, S. Bedathur	Computing n-Gram Statistics in MapReduce
MPI-I-2012-5-002	M. Dylla, I. Miliaraki, M. Theobald	Top-k Query Processing in Probabilistic Databases with Non-materialized Views
MPI-I-2012-5-001	P. Miettinen, J. Vreeken	MDL4BMF: Minimum Description Length for Boolean Matrix Factorization
MPI-I-2012-4-001	J. Kerber, M. Wand, M. Bokeloh, H. Seidel	Symmetry Detection in Large Scale City Scans