

# CATE: Context-Aware Timeline for Entity Illustration

Tran Anh Tuan  
Max Planck Institute  
for Informatics  
atran@mpi-inf.mpg.de

Shady Elbassuoni  
Max Planck Institute  
for Informatics  
elbass@mpi-inf.mpg.de

Nicoleta Preda  
Laboratoire PRISM  
Université de Versailles SQ  
preda@prism.uvsq.fr

Gerhard Weikum  
Max Planck Institute  
for Informatics  
weikum@mpi-inf.mpg.de

## ABSTRACT

Wikipedia has become one of the most authoritative information-sources on the Web. Each article in Wikipedia provides a portrait of a certain entity. However, such a portrait is far from complete. An informative portrait of an entity should also reveal the context the entity belongs to. For example, for a person, major historical, political and cultural events that coincide with her life are important and should be included in that person’s portrait. Similarly, the person’s interactions with other people are also important. All this information should be summarized and presented in an appealing and interactive visual interface that enables users to quickly scan the entity’s portrait.

We demonstrate CATE which is a system that utilizes Wikipedia to create a portrait of a given entity of interest. We provide a visualization tool that summarizes the important events related to the entity. The novelty of our approach lies in seeing the portrait of an entity in a broader context, synchronous with its time.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*retrieval models, search process*

## General Terms

Algorithms, Design

## Keywords

knowledge ranking, visualisation tools, timeline

## 1. INTRODUCTION

Wikipedia contains millions of articles about people, countries, historical events, research topics, inventions, etc. Typically, each article in Wikipedia describes an entity. Wikipedia contains also a hierarchy of categories, where each entity belongs to a set of categories, and each category is a sub-category of one or more categories. For example, the article about the famous German mathematician Carl Friedrich Gauss is included in the categories `1777_births`, `18th-century_mathematicians`, `German_mathematicians`, etc. The categories provide a *context* for the

entity. That is, for Gauss, we can infer from his categories that he is German, a mathematician, lived in the 18th century, etc.

The purpose of this work is to construct an extended timeline for a given entity that takes into consideration the several contexts the entity belongs to in order to provide a more complete and informative portrait of the entity. We believe that the contexts should play a role in the timeline construction for the following three reasons: 1) they represent an additional source of information that can be exploited to extract relevant events, 2) they provide a background for the entity and explain how it interacts with other entities within its contexts, and 3) they serve as a means of enabling users to focus on events related to one or more contexts of interest.

The timeline of Gauss shown in Figure 1 contains the information that he was born in Braunschweig and that he studied in the University of Göttingen. In addition, it contains the event that the French revolution broke in 1789 which is a major historical event in Europe during his lifetime. The timeline also contains events related to the entities Legendre and Riemann, two famous mathematicians whose work is closely related to that of Gauss in the fields of number theory and differential geometry, respectively. All these events are retrieved from Gauss Wikipedia page or Wikipedia pages of other entities highly relevant to Gauss and his contexts.

In addition to the events related to the entity, CATE provides the user with a set of relevant contexts which are used to focus the timeline on one or more contexts of choice (as can be seen in the upper part of Figure 1). We define the context as an object with three attributes, namely time, space and topic. For example, for Gauss, these attribute are:

- time: 18-th century, 19-th century, ...
- space: Braunschweig, Brunswick, Germany, Europe, ...
- topic: number theory, differential geometry, astronomy, ...

In order to construct an informative timeline such as the one in Figure 1, we need to perform three main tasks. The first is to associate each entity with a set of contexts. Second, given an entity (and possibly a subset of its contexts), we need to retrieve relevant entities to such entity or its contexts. For example, for Gauss, the relevant entities are Legendre, Riemann, French revolution, etc. Finally, given the entity and its relevant contexts and entities, we need to extract the related events to place them on the timeline. CATE relies on Wikipedia as the source of information to perform all three tasks and we explain how in the rest of the paper.

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.

WWW 2011, March 28–April 1, 2011, Hyderabad, India.  
ACM 978-1-4503-0637-9/11/03.

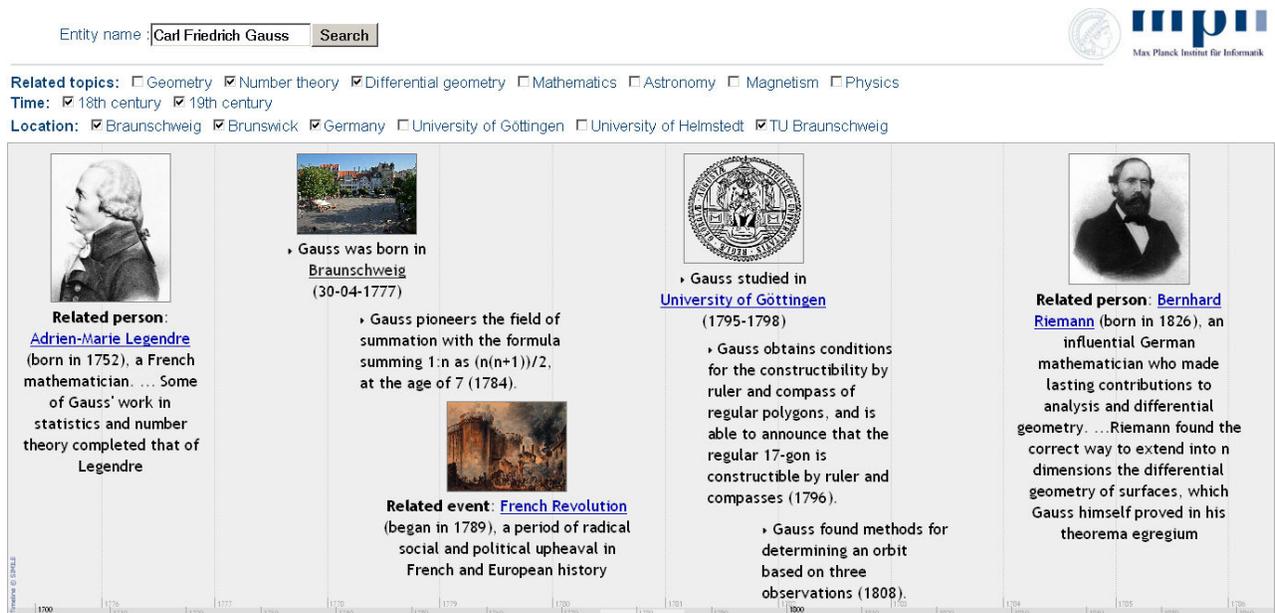


Figure 1: GUI Snapshot

## 2. SYSTEM ARCHITECTURE

As shown in Figure 2, CATE consists of five main components: a graphical user-interface (GUI), a retrieval engine, an event-description extractor, a data store, and an information-extraction tool. In a nutshell, CATE works as follows. The information-extraction tool is used to populate our data store and uses two sources of information: the Wikipedia corpus and its dumps, and the Web. To interact with CATE, the user inputs the name of an entity into a text box in the GUI. The GUI passes the input entity to the retrieval engine which retrieves all relevant contexts and entities from the data store. These are further sent to the event-description extractor, which extracts all related textual descriptions of the events. Finally, the results are sent to the GUI to be output to the user. We now explain each component in more details.

**GUI.** The GUI consists of 2 main sub-components. The first sub-component is the timeline illustrator where the events are positioned on the timeline and illustrated with images and text snippets. The second sub-component is the context selector (upper part of Figure 1) where relevant context-attributes are shown in the form of a menu. We do not enumerate the set of relevant contexts as many of them are overlapping. The user can use the context selector to control the visualized information on the timeline by selecting contexts of interest based on their attributes.

**Retrieval Engine.** The retrieval engine performs three types of retrieval tasks. It retrieves the most relevant contexts given an entity (e.g., number\_theory and differential\_geometry for Gauss). It also retrieves the most relevant entities given a certain context (e.g., the entity French revolution given the context 18th\_century\_Europe). In addition, given an entity and context, it retrieves the most relevant entities to the given entity and context (e.g., Riemann given the entity Gauss and the context differential\_geometry). We describe the ranking model used by the retrieval engine in Section 5.

**Event-Description Extractor.** The event-description extractor takes as input a Wikipedia article identifier and a query and retrieves the set of events related to the query from the article. The query can be either an entity name, a context name or both. The

output of the event-description extractor is a set of events in the form of an image, a text snippet and a timestamp.

**Information-Extraction Tool.** CATE’s extraction tool uses two sources of information: the Wikipedia dump and the Web. The Wikipedia dump is used to extract context information as well as hyperlink and text information which is used by both the retrieval engine and the event-description extractor. We explain our extraction algorithms in Section 3. The Web corpus on the other hand is used, via a well-known Web search-engine, to extract images.

**Data Store.** Our data store contains 3 databases. The first is the YAGO knowledge-base, an RDF-database that contains context information. An RDF-triple contains subject-property-object fields such as (Gauss, bornIn, Braunschweig). In entity-relationship style, subjects and objects are entities and properties represent relationships between entities. Hence, we can view YAGO as semantic graph where the nodes are entities and the labeled edges represent relationships. YAGO [6] has inferred class memberships from Wikipedia category names, and has integrated this information with the taxonomic backbone of WordNet. We extend the YAGO database with a new class of entities, namely the contexts.

Conceptually, a context  $C$  is an object with three types of attributes: time, space and topic. For example, for Gauss the following facts about his context are stored in the RDF-database:

Gauss inContext C		
C time 18th-century	C space Germany	C topic Mathematics
C time 19th-century	C space Braunschweig	C topic Physics
		C topic Astronomy

The attributes are mapped to YAGO entities. Hence, relationships between attributes are implicitly present in the database. For instance, for the space attribute we have the relationship partOf (e.g., (France, partOf, Europe)). The relationships between contexts are characterized based on the relationships between their attributes.

The second database in our data store is a text database. It contains 3 types of indices that are used by different components in

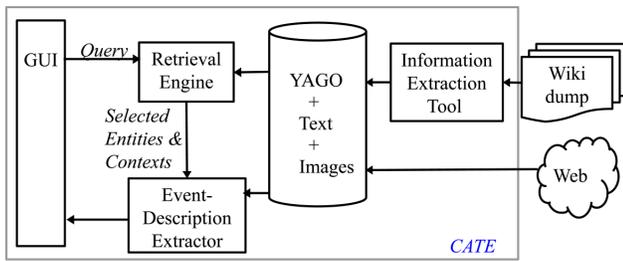


Figure 2: System Architecture

CATE. The first index is an inverted index over the hyperlinks. In particular, for each Wikipedia article, it stores all other articles that have outgoing links to it and the number of such links. The second index in the text database is a traditional inverted index that stores for each term all the articles that contains the term and the term frequency in the articles. These two indices are used by the retrieval engine to rank entities and contexts as we describe in Section 5. The third index in the text database is a full-text index that simply stores the full text of the articles. This is used by the event-description extractor to extract the snippets of the events which is explained in Section 6. Note that YAGO keeps a mapping between the entity and the corresponding Wikipedia article (if any exists). This is used by CATE to connect entities to articles.

The third database in our data store is an image database. It contains an image for each entity in YAGO.

### 3. INFORMATION EXTRACTION

Our extraction tool extracts three types of information: the context information, text and hyperlink information, and images.

**Context Extraction.** We utilize the Wikipedia categories and their hierarchy to extract contexts. Wikipedia category-names are usually composite names such as *18-th\_century\_mathematicians* or *German\_mathematicians*. These composite names consisting of orthogonal attributes pose a problem especially when constructing ontologies [6, 1]. However, looking closely, the category names in Wikipedia usually follow patterns that combine time, space and topic attribute-values. This observation was the basis of our 3-attributes model for contexts.

CATE extracts context information through a two-phase algorithm. In the first phase, it enumerates all Wikipedia categories and generates possible vocabularies for each of the three attributes. The vocabularies for time and space are extracted based on defined patterns, while those for topic are extracted using a set of automatically-learned patterns. In the second phase, the algorithm annotates each category using the extracted vocabularies. For example, the category *18-th\_century\_mathematicians* is associated with the attributes: time equals 18-th century and topic equals mathematics.

**Hyperlink and Text Extraction.** The Wikipedia dump provides a table with link information for each article. We utilize this and the full text of the articles to create the three inverted indices in the text database we explained in Section 2.

**Image Extraction.** In CATE, images are extracted from the Web using a Web search-engine. Selecting a relevant image for an entity is not so trivial as some entity names can be ambiguous. However, since this is not the main focus of our work, we use existing tools for resolving such cases such as [7].

## 4. ASSIGNING ENTITIES TO CONTEXTS

So far we have explained how to extract context information, and in this section we explain how we associate entities with the extracted contexts. Typically, each page in Wikipedia is annotated by users with the most relevant categories to which it belongs. This serves as the initial set of contexts for a given entity after setting their attributes values as we described in Section 3. However, Wikipedia categories alone offer incomplete and imprecise information. For instance, Gauss was included in the category *German\_physicists* which is a very broad context. By reading the contents of Gauss' Wikipedia page, one can realize that the majority of his contributions in physics are in the area of electro-magnetism. We explain two approaches for assigning entities to additional contexts next.

**Hyperlink Based Assignment.** This approach is based on hyperlink analysis. We assign an entity  $e$  to context  $C$  if the majority of entities that  $e$  links to, or the majority of entities that link to  $e$  belong to  $C$  as well. For example, consider the entity Gauss and the context *electro-magnetism* which is not one of Gauss' Wikipedia categories, and hence would not be considered as one of his contexts. However, it is very intuitive that Gauss *should* belong to this context if the article of Gauss contains many links to or from other entities that belong to the context *electro-magnetism*.

**Attribute Based Assignment.** The hyperlink based method would only identify contexts that are part of the Wikipedia-category hierarchy. However, we can combine attributes from different contexts in order to generate new contexts. For example, given that Gauss belongs to the contexts *18th\_century\_mathematicians* and *german\_mathematicians*, we can generate the new context *18th\_century\_Germany*. In addition, we can utilize the categories hierarchy and external ontologies such as WordNet and Geo-Names to generate further contexts such as *18th\_century\_Europe* using the fact that Germany is part of Europe from such ontologies. Actually, this is crucial for the inclusion of the entity French revolution in the timeline of Gauss because it is relevant (according to the Wikipedia contributors) not only to France but to whole Europe.

## 5. RANKING MODEL

We have three intermingling ranking problems: 1) ranking the contexts given an entity, 2) ranking entities given a context and an entity, and 3) ranking entities given a certain context only. For all three problems, we adopt a statistical-language-modeling approach [5], and we utilize our text database with its three indices to estimate the parameters of our model.

**Basic Setting.** We adopt the following notation. Each context  $C$  is associated with a set  $E(C) = \{e_1, e_2, \dots, e_n\}$  which is the set of entities that belong to context  $C$ . Additionally, each entity  $e_i$  is associated with a document  $D(e_i)$  which represents the Wikipedia article of  $e_i$ . For each such document, we construct a language-model (LM) which is a probability distribution over all the entities. We denote the parameters of the LMs as  $P(e|D(e_i))$  which is the probability of generating the entity  $e$  given the LM of document  $D(e_i)$ . This probability is estimated using a maximum-likelihood estimator after employing Dirichlet smoothing as in most common LM approaches [10]. The maximum-likelihood estimator can be computed in various ways and we experiment with two different methods. The first method uses the number of links in  $D(e_i)$  to the article of entity  $e$  to estimate the probability  $P(e|D(e_i))$  and the second method uses the text of the document  $D(e_i)$ . The latter is motivated by the observation that there are many missing links in Wikipedia. Note that having a link to an entity in some Wikipedia

article is a much stronger evidence that this entity is relevant to that article as opposed to just being mentioned in the text.

**Ranking Contexts.** Given an entity  $e$  and the set of contexts it belongs to, we rank the contexts based on their probabilities of generating the entity  $e$ . We estimate this probability by constructing a language-model for each context  $C$  as a mixture model over the documents of its entities as follows:

$$P(e|C) = \frac{1}{n} \sum_{i=1}^n P(e|D(e_i)) \quad (1)$$

where  $P(e|D(e_i))$  is the probability of generating entity  $e$  given the document of entity  $e_i$  which can be estimated as described in the beginning of this section.

**Ranking Entities Relevant to a Given Entity within a Context.**

The second ranking problem we have is to retrieve the most relevant entities given an entity  $e$  and a context  $C$ . We rank the entities based on their probability of being generated given  $e$  and  $C$  which we denote by  $P(e'|C, e)$ . To compute such probability, we construct a language model for  $C$  as a mixture model of the documents LMs of all entities in  $C$ . However, we only restrict this to documents that contain  $e$  as well to accommodate for the conditioning over  $e$ . That is, we ignore the documents of entities that belong to  $C$  and do not contain  $e$ . To this end, let the set of documents of entities that belong to  $C$  and contain  $e$  be  $\{D(e_1), D(e_2), \dots, D(e_l)\}$ . This way, the probability  $P(e'|C, e)$  is equal to:

$$P(e'|C, e) = \frac{1}{l} \sum_{i=1}^l P(e'|D(e_i)) \quad (2)$$

**Ranking Entities within a Context.** The third and final ranking problem we deal with is ranking entities  $e'$  that belong to a certain context  $C$ . This can be easily done using the probability  $P(e'|C)$  which is computed as described in Equation 1.

## 6. EXTRACTING EVENTS

Our event-description extractor takes as an input a query and a Wikipedia article, and retrieves the top- $k$  events associated with the query from the article. The query can be either an entity name, a context name or both. An event is a text snippet, a timestamp and an image. The algorithm works as follows. First, we identify from the article all snippets  $S$  that contain time expressions. This can be easily done using tools such as [2]. Next, we rank the identified snippets based on their probability of generating the query  $P(Q|S)$  (in line with our ranking model of contexts and entities) and output the highest-ranked  $k$  snippets.

Finally, to associate an image to the event which is used for illustration on the timeline, we identify the main entity the event is about, and then retrieve the image associated with that entity from our image database.

## 7. DEMONSTRATION

For the demonstration, we will provide two interfaces to interact with CATE: a user interface where the timeline is shown, and a developer interface that demonstrates the different algorithms CATE uses for context assignment and ranking.

In the developer interface the user can explore the hierarchy of contexts. Given an entity of interest, the user is presented with the Wikipedia categories the entity belongs to. The developer interface also shows the set of contexts that CATE assigns to the entity using the two algorithms briefly described in Section 4. In addition to context assignment, we also show the ranking algorithms at work.

In particular, we show the output of our three ranking problems that we highlighted in Section 5 for a given entity.

## 8. RELATED WORK

Our work touches various subjects in the area of knowledge discovery. The closest to our work is the Correlator system [9]. The Correlator system also operates on top of Wikipedia. It is mainly used to correlate a set of entities, and derives relationships between them based on the Wikipedia articles and categories that contain them. The Correlator contains a timeline component as well. However, the Correlator timeline is based only on the article of the given entity, and does not leverage context information as we do in CATE in order to provide a more complete and comprehensive portrait of the entity. Another related work is the E-Culture demo [4] which is a multimedia faceted-search tool that uses annotations from different data sources including WordNet. Given a certain category or class, the system retrieves a set of entities (e.g., art works or artists) that are most relevant to the given category, and shows the evolution of the category over the timeline. This is conceptually different from CATE whose goal is to provide a timeline for a given entity.

Extracting events has been addressed in the area of information extraction to some extent. Recently, the authors in [8] introduced an information extraction framework for annotating database relationships with time information. We see such work as complementary to ours and that our algorithms and system can make use of. Furthermore, the emergence of such works strongly confirms the interest that people have in anchoring knowledge in the time and space dimensions.

Wikipedia categories reflect user perspectives of categorizing an article into many different concepts, and is a goldmine for extracting ontologies. Recent work such as [1, 3] addressed tapping Wikipedia categories for knowledge acquisition.

## 9. REFERENCES

- [1] V. Nastase and M. Strube. Decoding wikipedia categories for knowledge acquisition. In *AAAI*, 2008.
- [2] J. Pustejovsky, J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, and G. Katz. Timeml: Robust specification of event and temporal expressions in text. In *IWCS*, 2003.
- [3] P. Schönhofen. Identifying document topics using the wikipedia category network. *Web Intelligence and Agent Systems*, 7(2):195–207, 2009.
- [4] G. Schreiber, A. Amin, M. van Assem, V. de Boer, L. Hardman, M. Hildebrand, Z. H. Laura Hollink, J. van Kersen, and M. de Niet. Multimedial e-culture demonstrator. In *ISWC*, 2006.
- [5] F. Song and W. B. Croft. A general language model for information retrieval. In *SIGIR*, 1999.
- [6] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A Core of Semantic Knowledge. In *WWW*, 2007.
- [7] B. Taneva, M. Kacimi, and G. Weikum. Gathering and ranking photos of named entities with high precision, high recall, and diversity. In *WSDM*, 2010.
- [8] Y. Wang, M. Zhu, L. Qu, M. Spaniol, and G. Weikum. Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In *EDBT*, 2010.
- [9] H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, and G. Attardi. Ranking very many typed entities on wikipedia. In *CIKM*, 2007.
- [10] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, 2001.