Terms in Time and Times in Context: A Graph-based Term-Time Ranking Model

Andreas Spitz, Jannik Strötgen, Thomas Bögel, Michael Gertz
Institute of Computer Science, Heidelberg University
Im Neuenheimer Feld 348, 69120 Heidelberg, Germany
{spitz,stroetgen,boegel,gertz}@informatik.uni-heidelberg.de

ABSTRACT

Approaches in support of the extraction and exploration of temporal information in documents provide an important ingredient in many of today's frameworks for text analysis. Methods range from basic techniques, primarily the extraction of temporal expressions and events from documents, to more sophisticated approaches such as ranking of documents with respect to their temporal relevance to some query term or the construction of timelines. Almost all of these approaches operate on the document level, that is, for a collection of documents a timeline is extracted or a ranked list of documents is returned for a temporal query term.

In this paper, we present an approach to characterize individual dates, which can be of different granularities, and terms. Given a query date, a ranked list of terms is determined that are highly relevant for that date and best summarize the date. Analogously, for a query term, a ranked list of dates is determined that best characterize the term. Focusing on just dates and single terms as they occur in documents provides a fine-grained query and exploration method for document collections. Our approach is based on a weighted bipartite graph representing the co-occurrences of time expressions and terms in a collection of documents. We present different measures to obtain a ranked list of dates and terms for a query term and date, respectively. Our experiments and evaluation using Wikipedia as a document collection show that our approach provides an effective means in support of date and temporal term summarization.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*

Keywords

Temporal information, time-based analysis, ranking

1. INTRODUCTION

What comes to your mind when you think of the year 1492? What dates do you think of when you hear the name "Kennedy"? Obviously, the answers to these questions depend on one's individual knowledge. However, given a large corpus of documents such as Wikipedia, one might expect that by analyzing documents, especially with respect to the temporal information they contain, it is possible to determine terms and dates, respectively, that should be indicative for answers to questions such as the above.

There are several approaches that exploit temporal information extracted from documents for the purpose of text analysis and exploration. These include, among others, the extraction of temporal facts and events from Wikipedia [10] or the construction and exploration of timelines, e.g., for popular events [2], temporal clustering of documents [1], or the summarization of events with respect to an entity [18]. In particular the summarization aspect is appealing as it provides a means to obtain a succinct description of an entity using text from large corpora. This type of summarization can also be found in the context of geographic information, for example, for place summarization where a set of terms is associated with an entity of type geography or location, see, e.g., [13]. Common to all these approaches is that they exploit the co-occurrence of terms with entity types.

In this paper, we present a new approach that, given a date of some granularity such as a day, a month, or a year, determines a set of terms that are relevant for that date and describe that date. Key to this approach is the extraction and normalization of temporal expressions found in documents using state-of-the-art temporal taggers such as HeidelTime [17]. Analogous to the above approach, we also show how to associate a ranked list of dates with a query term. For example, for a query term such as "Kennedy", one would then obtain a ranked list of dates that are related to that term, such as when he took office or the day of his assassination. The hypothesis is that the more often a term appears with a temporal expression in a document, typically at the sentence level, the more likely the term and date are related. These two techniques can easily be extended to associate a list of dates with a query date, meaning that the context of the ranked dates is the same or similar to the query date. This way, dates of similar events or recurring dates can be determined.

The foundation of our approach is the representation of dates of different granularities and terms as a weighted bipartite graph. To determine the relevance of a date for a term (and vice versa), we employ a technique similar to col-

laborative filtering used in recommender systems [11]. We apply different versions of the *tf-idf* scheme and cosine similarity to determine the relevance and thus ranking of terms for a date and dates for a term, as well as rankings within terms and dates, respectively. Our experiments and evaluations using the English Wikipedia corpus demonstrate the utility of the proposed techniques in support of temporal information retrieval. For some specific use-cases, our approach, though simple in its realization, shows extremely good results, especially when exploring event-like concepts.

The remainder of the paper is structured as follows. After a brief review of related work, we present our term/time graph model in Section 3. In Section 4, we present several experimental results and evaluate the meaningfulness of the results obtained by our approach. In Section 5, we summarize the paper and outline some ongoing work.

2. RELATED WORK

Temporal information is prevalent in many documents across domains and provides a method of inducing structure to unstructured document collections due to the underlying ordering aspects of time. Thus, exploiting temporal information extracted from documents has become an important part of information retrieval [3, 4]. In the following, we present related approaches tackling time-based representation and exploration of document collections from different angles and for different domains.

Alonso and Shiells present an approach for constructing summary-style timelines for scheduled events by analyzing Twitter data about respective events [2]. Using a frequency-based approach, the relevance of terms is determined per time point, and timelines are enriched with most relevant keywords for summarizing the evolution of events. While this approach is suitable in the context of micro-blogging and specific events due to the limited number of characters per post, frequency in isolation does not capture the importance of individual terms in large document collections.

Using a news archive, Setty et al. generate timelines highlighting important dates for a specific user query (e.g., about persons or events) [15]. Importance is measured on the assumption that the top-k time-travel query result for the topic of interest change significantly at important times. Thus, their approach exploits the document creation times of news articles. In contrast, our approach uses a document collection and exploits the temporal information available in the unstructured content of the documents. In addition, our approach takes into account different granularities of date expressions and can thus be used to analyze document collections at different temporal granularities.

Jatowt et al. use a time-term association graph in an estimation of the focus time of entire documents [8]. In contrast to our approach, this graph is constructed from external information extracted from a corpus of news articles. Based on this association information, they identify discriminative time-term associations and employ these to estimate the focus time of given documents. The method is evaluated on a selection of Wikipedia articles for which a unique focus time can be defined. In an approach that is also based on an external knowledge base of news articles, Gupta and Berberich identify time intervals of interest for given keyword queries based on pseudo-relevant documents [7]. They employ a probabilistic approach for the selection of suitable

documents for a given query and generate a time interval from the contained temporal expressions.

Kanhabua and Nejdl analyze temporal anchor texts extracted from Wikipedia's edit history to track and detect the evolution of entities and events [9]. As in the approaches described above, temporal metadata (in this case, the edit history) is used to discover time-related knowledge. In contrast, Filannino and Nenadic suggest to create temporal footprints from single Wikipedia pages describing concepts such as persons [6]. Given a Wikipedia article, temporal expressions extracted from the article's text are used for designing a temporal footprint of the concept described by the article. For a given set of templates, Kuzey and Weikum enrich facts and events with temporal information by extracting temporal information about them from Wikipedia documents by applying a set of templates for facts and events [10]. While these approaches are similar to our work since temporal information described in the documents' text is exploited, we do not analyze single Wikipedia pages or use concept templates but rather study the relationship between dates and content (e.g., entities, events, keywords) in a general and global way. For this, we consider the full document collection at once without limiting our knowledge extraction to specific concepts.

In addition, our approach does not only determine the most relevant times for a concept but also vice-versa, i.e., the most relevant keywords for a specific time point, and even allows for a comparison of dates and terms. Thus, our approach can be considered as a two-way summarization approach for points in time and concepts.

3. MODEL

In this section, we first explain the characteristics of our graph representation. Then, we present our ranking functions to determine relevant terms for dates and vice-versa.

3.1 Time-Term-Model

Graph structures enable an efficient representation of relations between entities with respect to space and time complexity of the performed analysis. Furthermore, they provide a well defined framework for an analysis of such relations. Bipartite graphs in particular are well suited for such a task when the set of entities can be partitioned into two distinct sets and identical operations are of interest for both. Therefore, we select such a representation of the date-term co-occurrence data that we extract from the sentences of the data set and, based on this, define ranking functions that are equally applicable to both dates and terms in the graph.

Let T be a set of terms (words) and D a set of dates with differing granularity (days, months, and years). Then a simplified set of sentences in a bag-of-words model can be defined as $S \subseteq (T \cup D)^*$, i.e., a set of sets that contain both terms and dates with no particular ordering. Based on this, we define an undirected bipartite co-occurrence graph of terms and dates $G := (V, E)$ with $V := T \cup D$ and $E \subseteq T \times D$, in which an edge $e = (t, d) \in E$ iff there exists a sentence $s \in S$ such that $d \in s$ and $t \in s$, i.e., the nodes that represent a term and a date are connected in the graph if there exists at least one sentence in which they co-occur. We further define a weight function $\omega : E \to \mathbb{N}$ for the edges where $\omega(t, d) := |\{s \in S : t \in S \wedge d \in S\}|$, i.e., the number of co-occurrences of nodes d and t. The weighted, bipartite adjacency matrix of G with dimension $|D| \times |T|$ is

then defined as $A_{ij} := \omega(d_i, t_j)$ iff $(d_i, t_j) \in E$ and $A_{ij} := 0$ otherwise. For any date d_i, the adjacency vector $\vec{d_i}$ then equals the i-th row vector of A. The adjacency vectors of terms are defined analogously as column vectors. We call the number of edges that are adjacent to a node the degree and write $deg(t)$ or $deg(d)$.

To account for the differences in the granularity of dates, we consider them to be sets of time points for which an inclusion hierarchy exists. That is, for each day there exists a month in which it is included and the same relation holds between months and years. As a result, we partition D into three subsets $D = D_y \cup D_m \cup D_d$. This hierarchy can be reflected in the graph by requiring that for all $(d, t) \in E$, if there exists a $d' \in D$ such that $d \subset d'$, then we also have $(d', t) \in E$ and $\omega(d', t) \geq \omega(d, t)$, e.g., an edge between a day and a term also induces an edge between this term and both the month and year that include the day. The weight of these induced edges cannot be smaller than that of the original edge.

3.2 Ranking Functions

Based on this graph, we now introduce ranking functions to obtain ranked lists of all terms and dates for a given term or date query. We define such a ranking function simply as $r_{XY} : X \to \mathbb{R}^{|Y|}$ where $X, Y \in \{D, T\}$, i.e., we map a node from either set of the graph to a vector of ranking scores with a dimension equal to the target set. As a result, we obtain two different types of ranking functions: Heterogeneous functions r_{XY}, which map a node from one set to scores for the opposite set, and homogeneous functions r_{XX}, which map a node to scores for nodes of the same set.

An established approach to instantiate the heterogeneous ranking function would be the use of a point-wise mutual information measure. Since it has been shown that the inherent independence assumption of mutual information implies an overly simplistic model for a bipartite graphs [21], we employ a different instantiation. We observe that by creating so-called *temporal term profiles* (or inversely, *contextual date profiles*), we intend to equate dates with documents that contain the terms they co-occur with. As a measure of the importance of a co-occurrence between a specific date and a term, we build upon the analogy to terms that are contained in a document and use a *tf-idf* score. We then observe that, in the graph, the weight $\omega(d, t)$ is equivalent to the number of common occurrences of d and t in the data, meaning that it represents a kind of term frequency (tf). Similarly, the degree of a term (i.e., the number of adjacent edges) is equivalent to the frequency with which it appears alongside a particular date. As such, it is a kind of document frequency, and we can compute an inverse document frequency (idf) based on this. By combining the two, we arrive at a version of the *tf-idf* score that is adapted to the graph representation:

$$tf\text{-}idf(d, t) := \omega(d, t) \log \frac{|D|}{deg(t)}.$$

Since there is no difference between the two sides of the graph from a graph theoretic point of view, we may also exchange terms for dates and arrive at a rating of the importance of dates for a given term

$$tf\text{-}idf(t, d) := \omega(d, t) \log \frac{|T|}{deg(d)}.$$

Ranking functions are then given simply as the *tf-idf* scores of all terms (dates) with respect to the query date (term).

When we consider possibilities for a homogeneous ranking function, we essentially rank the importance of connections between nodes in the same set. This amounts to a so-called *one-mode projection* of the bipartite graph onto one of the two sets of nodes by introducing new edges between nodes in the same set if they share at least one neighbor in the original graph. This process introduces a great number of new edges, thus making the generation of weights that can be used for ranking the edges of the projection a crucial element of the process. While there exist very precise, statistically motivated approaches as well as available toolkits for this task that are based on the exact structure of the bipartite graph [21, 16], these methods are prohibitively expensive for a data set as large as Wikipedia. In analogy to document similarities in the field of text mining [14] and to bipartite item-user graphs in collaborative filtering for product recommendation [11], we thus use a cosine similarity of the adjacency vectors associated with the nodes as a measure of similarity:

$$cos(t_i, t_j) := \frac{\vec{t_i} \circ \vec{t_j}}{\|\vec{t_i}\|\|\vec{t_j}\|}.$$

Here, $\circ$ denotes the scalar product and $\| \cdot \|$ the euclidean norm. Again, the same task can obviously be performed for the second side of the graph to arrive at a measure of similarity between dates and the ranking function is then given by the results of the pairwise cosine of adjacency vectors between the query and all other nodes in the same set.

Based on these functions, we are now able to create rankings of closely related terms and dates for both dates and terms, allowing us to extract summarizations of either kind from the data and uncover event-like connections. Even though the functions themselves are quite simple, their performance is astonishing as we show in the following.

4. EVALUATION

In this section, we first describe the document collection and processing as well as the experimental setup. Then, our evaluation and experiments demonstrate the value of our approach.

4.1 Document Processing

To create a bipartite graph for dates and terms, we run the following strategy. Given a set of documents (the full English Wikipedia dump), we apply a temporal tagger to extract and normalize temporal expressions occurring in the documents' texts. For the task of temporal tagging, we use HeidelTime [17] since it is a domain-sensitive temporal tagger and not limited to processing only news-style documents. In addition, its extraction and normalization quality was proven to be sophisticated in the context of several evaluation campaigns [19, 20].

Note that a temporal tagger usually extracts temporal expressions of four types, as distinguished in the temporal markup language TimeML [12]: dates, times, durations, and set expressions. For our analysis, we focus on temporal expressions of types time and date, and ignore expressions of granularities coarser than year since these are too unspecific for a meaningful analysis. In addition, we map all temporal expressions that refer to a point in time smaller than gran-
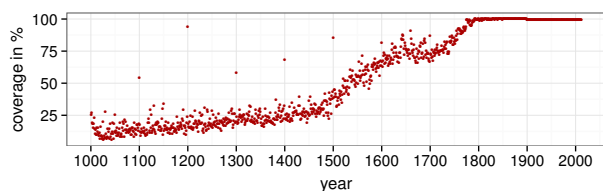
Figure 1: Coverage of dates as the percentage of possible dates by year that occur in the data set.

ularity day, month, and year to the respective day, month, and year to guarantee a clean hierarchy.

In the next step, we create pairs of temporal expressions and terms that co-occur with the respective expressions in the same sentence. Thus, we create the *temporal term profiles* and *contextual date profiles* (cf. Section 3.2), which are jointly stored in a bipartite graph. To avoid the inclusion of uncharacteristic information and an unnecessary expansion of our graph, we exclude stop words. To ensure the hierarchical property of dates in the resulting graph, for each term that is connected to a node representing a day or month, we also include an edge to the node(s) representing the encompassing date(s) and weight it accordingly.

While a total of 3,079,620 Wikipedia articles contain at least one temporal expressions and thus contribute to the data set, more than 29 million temporal expressions are extracted. The resulting graph consists of 3,748,730 terms, 210,375 (unique) dates and 110,639,525 edges connecting them. As we show in Figure 1, the number of dates that are included in the data set is good beginning with the 16th century and perfect for dates beyond 1800.

4.2 Experimental Setup

After the computation of co-occurrences, we store the entire bipartite graph in memory in a dual adjacency-list based structure for both sets of nodes to allow for efficient query processing. Adjacency information, edge weights and pre-computed degrees of nodes can be stored in 4GB of memory in a Java implementation. On an Intel Core i7 CPU with four cores and 16GB memory running Ubuntu 12.04, the computation of *tf-idf* scores for a query is possible in near real-time. The computation of cosine ranking scales with the degree of the query node and is equally fast for dates and terms with medium degree. This increases up to a run-time of minutes for the most frequent terms, although this is trivially parallelizable and scales with the number of cores.

4.3 Experiments

To provide an overview of the capabilities of our approach, we perform experiments for the different granularities of temporal expressions that are present in the data set. We primarily focus on days and months, since we find years to be too broad a category to obtain meaningful results in most contexts. We begin with a selection of exemplary date and term queries that show the strengths and drawbacks of our approach. We then provide an evaluation of the method by using a ground truth of cyclical dates for queries and finally present an example of a possible application.

4.3.1 Experimental Results

To give an example of a query for terms in the graph, we provide the results for a query of "kennedy", which yields

the list of dates *1963, 1961, 1960, 1963-11, 1962* and *1963-11-22*. Here, November 22, 1963 is the date of John F. Kennedy's assassination, 1960 the year he was elected president, 1961 the year he began his presidency and 1962 the year of the Cuban Missile Crisis. Thus, we are not only able to identify the years of Kennedy's presidency, but we also find the probably best known event during this time frame.

With time expressions, we have more options due to the granularity. For the date query "1492", we find as top ranked terms (in this order) *jews, columbus, spain, granada, expulsion, spanish, christopher* and *americas*. This coincides with the discovery of America by Columbus and the Expulsion of the Jews from Spain. To give another example, a query for "1990-10-03", the day of the German Reunification, yields the top ranked terms *reunification, germany, german, gdr, berlin, republic* and *east*. In general, we find that it is easier to precisely pinpoint events in smaller time frames, since a large number of similar, small events tends to drown out events that could be considered more important. An example of this is the month "2005-04", during which Pope Benedict XVI was elected. Here, we find the terms *released, album, announced, season, band* and *series* to be most prominent, indicating that the multitude of music and TV releases had a greater combined impact during that month than the election, and we do not find a mention of the term *pope* until position 94 in the list. This indicates that with Wikipedia as a data set, more elaborate filtering by category of the article may be required, if popular news are unwanted.

In Table 1, we present self-explanatory results for further example queries together with *tf-idf* values as well as frequency information.

4.3.2 Day-focused Evaluation

To demonstrate the effectiveness of our approach, we perform an evaluation of results for the ranking. We do this by selecting a periodic event that occurs once per year with a slight variation in the exact date, namely the election Day of the United States, which falls on the Tuesday after the first Monday in November. We conjecture that each of these dates should be similar with respect to their contextual date profiles. A ranking of similar dates by cosine similarity for any given Election Day should thus contain the dates of other Election Days in top positions. Obviously, the more such dates of other instances of this event are highly ranked, the better. We compile a list of Election days between the years 1848 and 2013 to use as ground truth. These can be split further into general Election Days, which occur every year, and the subset of presidential Election Days, which occur once every four years.

Based on this evaluation data, we compute predictions of the most similar dates by cosine similarity for all Election Day dates in the evaluation data set. We label the results as positive if a predicted date is also contained in the set of Election Days (presidential Election Days) and as negative if it is not. Precision is then computed as the ratio of positives to negatives among the top k predictions for a sliding window of $k = 1, ..., 100$. As shown in Figure 2, the precision of even this rough approach is quite high with 0.6 initially and still as high as 0.5 for the first 20 most similar dates. As expected, initial results for the presidential Election Days are slightly better yet decrease more sharply, since the number of such days is smaller.

"1942-12-07"	tf-idf	tf	df	"1215-06-15"	tf-idf	tf	df	"tsunami"	tf-idf	tf	df
pearl	3228.7	1009	8576	carta	79.7	14	709	2004	3097.2	1374	393475
harbor	2927.0	1020	11933	magna	71.2	14	1298	2011	2753.9	1313	460264
japanese	1829.5	721	16633	barons	46.9	10	1928	2011-03	1878.5	464	65407
attack	1569.3	714	23361	runnymede	40.5	6	247	2004-12-26	1658.0	238	3536
war	618.5	462	55150	king	20.4	12	38400	2011-03-11	1474.2	226	5508
attacked	516.0	194	14717	oaths	17.1	3	714	2005	1030.6	476	430107
hawaii	475.1	145	7945	king's	15.1	5	10200	2004-12	734.8	162	40186
states	413.3	306	54506	repudiation	13.6	2	231	2005-01	465.5	102	39062
united	403.7	314	58164	fealty	12.4	2	424	2006	301.7	147	481555
air	365.2	176	26420	john	11.8	11	71893	2010	295.2	148	510254

Table 1: Query results with *tf-idf* score, term / date frequency (*tf*) and document frequency *df*.
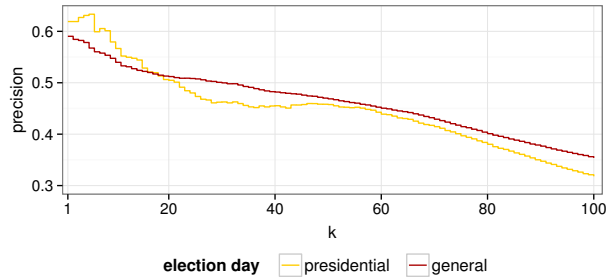

Figure 2: Average precision over all Election Days (red) and presidential Election Day (yellow) in the evaluation data set for the first k predictions by cosine similarity. For the general Election Day data set, all dates of Election Days are used for prediction and counted as positives. For the presidential Election Days, only Election Days in years divisible by four are used for prediction and counted as true positives. The results are averaged over 1000 samples to account for ties in the ranking by cosine.

An evaluation of the recall of this approach is less feasible, but in order to evaluate the overall positioning of similar dates, we also consider the Receiver-Operator Characteristic (ROC) [5] for the entire list of predictions. To obtain a single score for the performance of the measure on each Election Day, we compute the area under this curve (AUC). The results are shown in Figure 3 and, as one can see, the overall performance is excellent with very few exceptions at the end of the 19th century, which is possibly caused by the quick rise and demise of the People's Party during that period in American politics.

4.3.3 Month-focused Exploration

In our second experiment, we analyze the quality of contextual date profiles based on months during World War II by assigning an activity score to countries based on their relevance in the selected month as determined by our approach. Here, we compile a list of European countries during World War II and assign to each country c its name $t_n(c)$ and respective adjective $t_a(c)$. For example, the country of Italy would be assigned $t_n(c) = italy$ and $t_a(c) = italian$. For each month d during the period of World War II, we then compute a ranking of all terms by *tf-idf* score. From this, we derive an activity score for each country as the normalized sum of *tf-idf* values for the name and adjective, i.e.,

$$act(c, d) := \frac{\textit{tf-idf}(d, t_n(c)) + \textit{tf-idf}(d, t_a(c))}{\max[\textit{tf-idf}(d, \cdot)]}$$
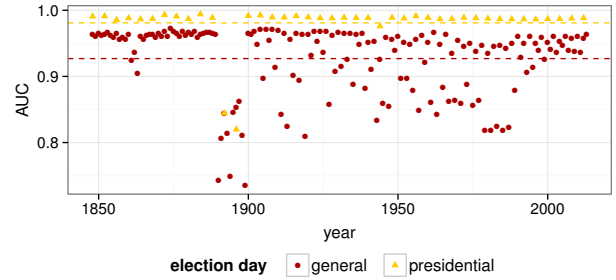

Figure 3: AUC scores for all $210,374$ predictions by cosine similarity for each Election Day in the evaluation data set, plotted against the year of the Election Day. Dashed lines denote the average AUC.

The activity thus gives a score in the interval $[0, 2]$, with higher values denoting a higher relevance of a country's activities during the chosen month. We plot the resulting activity scores as a heatmap for selected months of the war during which major events took place and provide a short summary of the event in question to highlight the relevance of the results in Figure 4. For all shown major events, we can observe a clear correlation between the participants of the event and their activity score. The only exception is the perhaps best known date of World War II, namely D-Day. This is likely due to the strong correlation between the date and the term *normandy* instead of *france*.

5. CONCLUSIONS & ONGOING WORK

In this paper, we presented an approach to obtain a ranked list of terms for individual dates of different granularities, as well as ranked dates for specific query terms. Modelling co-occurrences of time expressions and terms with a weighted bipartite graph facilitates experiments that are flexible with respect to the used relevance metric. We demonstrated the usefulness of our approach with real-world examples and evaluated the underlying methods of our ranking approach and measures based on the prediction of periodic events.

Currently, we are working on comparing term-time vectors extracted from Wikipedia dumps of different languages to study language-dependent differences. Furthermore, we focus on specific types of named entities such as persons and an evaluation of different ranking functions.
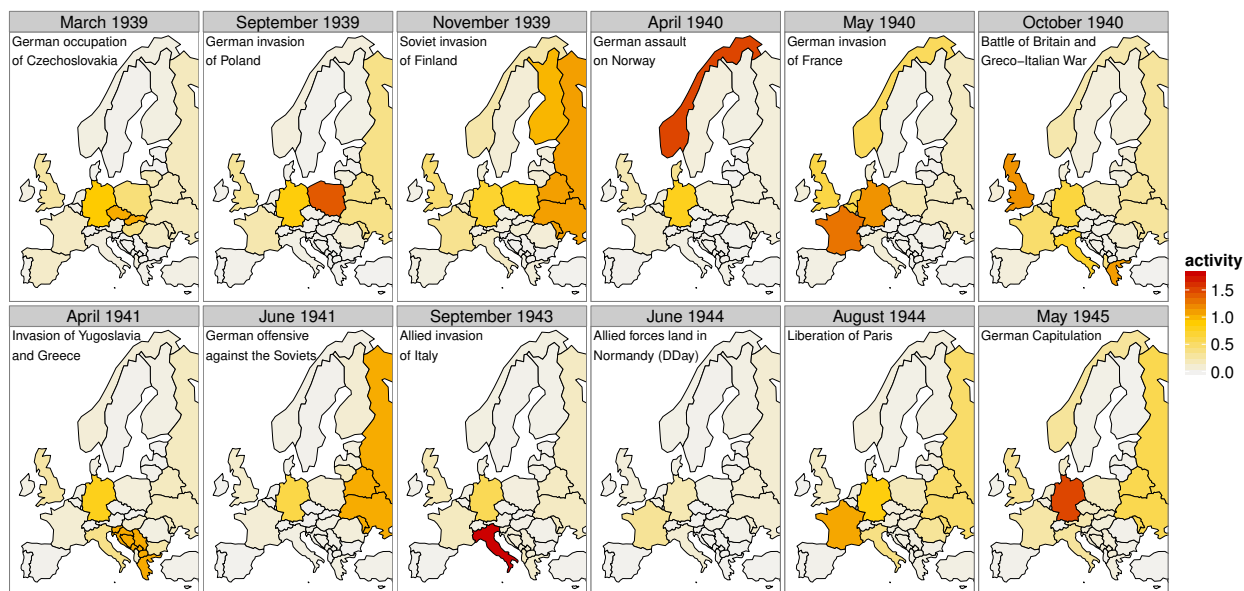
Acknowledgements

Figure 4: Overview of European countries for selected months in which major events took place during World War II. Countries are highlighted by the relevance of the *tf-idf* scores of the countries' name and adjective as described by the activity score. Note that for ease of recognition, modern country borders are used instead of contemporary borders, but countries are grouped accordingly (i.e., Yugoslavia, the Soviet Union and Czechoslovakia).

6. REFERENCES

[1] O. Alonso, M. Gertz, and R. A. Baeza-Yates. Clustering and Exploring Search Results using Timeline Constructions. In *CIKM'09*, pages 97–106, 2009.

[2] O. Alonso and K. Shiells. Timelines as Summaries of Popular Scheduled Events. In *WWW'13, Companion Volume*, pages 1037–1044, 2013.

[3] O. Alonso, J. Strötgen, R. Baeza-Yates, and M. Gertz. Temporal Information Retrieval: Challenges and Opportunities. In *TempWeb'11*, pages 1–8, 2011.

[4] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt. Survey of Temporal Information Retrieval and Related Applications. *ACM Computing Surveys*, 47(2):15:1–15:41, 2014.

[5] T. Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[6] M. Filannino and G. Nenadic. Mining Temporal Footprints from Wikipedia. In *Proceedings of the First AHA!-Workshop on Information Discovery in Text*, pages 7–13, 2014.

[7] D. Gupta and K. Berberich. Identifying Time Intervals of Interest to Queries. In *CIKM'14*, pages 1835–1838. ACM, 2014.

[8] A. Jatowt, C.-M. Au Yeung, and K. Tanaka. Estimating Document Focus Time. In *CIKM'13*, pages 2273–2278. ACM, 2013.

[9] N. Kanhabua and W. Nejdl. On the Value of Temporal Anchor Texts in Wikipedia. In *TAIA'14*, 2014.

[10] E. Kuzey and G. Weikum. Extraction of Temporal Facts and Events from Wikipedia. In *TempWeb'12*, TempWeb '12, pages 25–32, 2012.

[11] G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-item Collaborative Filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[12] J. Pustejovsky, R. Knippen, J. Littman, and R. Saurí. Temporal and Event Information in Natural Language Text. *Lang Resour Eval*, 39(2-3):123–164, 2005.

[13] T. Rattenbury, N. Good, and M. Naaman. Towards Automatic Extraction of Event and Place Semantics from Flickr Tags. In *SIGIR'07*, pages 103–110, 2007.

[14] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. 1983.

[15] V. Setty, S. Bedathur, K. Berberich, and G. Weikum. InZeit: Efficiently Identifying Insightful Time Points. In *VLDB'10*, pages 1605–1608, 2010.

[16] A. Spitz, K. A. Zweig, and E.-Á. Horvát. SICOP: Identifying Significant Co-interaction Patterns. *Bioinformatics*, 29(19):2503–2504, 2013.

[17] J. Strötgen and M. Gertz. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013.

[18] T. A. Tuan, S. Elbassuoni, N. Preda, and G. Weikum. CATE: Context-aware Timeline for Entity Illustration. In *WWW'11*, pages 269–272, 2011.

[19] N. UzZaman, H. Llorens, L. Derczynski, J. F. Allen, M. Verhagen, and J. Pustejovsky. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *SemEval'13*, 2013.

[20] M. Verhagen, R. Saurí, T. Caselli, and J. Pustejovsky. SemEval-2010 Task 13: TempEval-2. In *SemEval'10*, pages 57–62, 2010.

[21] K. A. Zweig and M. Kaufmann. A Systematic Approach to the One-Mode Projection of Bipartite Graphs. *Soc Netw Anal Min*, 1(3):187–218, 2011.