Important Events in the Past, Present, and Future

Abdalghani Abujabal Klaus Berberich
Max Planck Institute for Informatics
Saarbrücken, Germany
{abujabal, kberberi}@mpi-inf.mpg.de

ABSTRACT

We address the problem of identifying important events in the past, present, and future from semantically-annotated large-scale document collections. Semantic annotations that we consider are named entities (e.g., persons, locations, organizations) and temporal expressions (e.g., during the 1990s). More specifically, for a given time period of interest, our objective is to identify, rank, and describe important events that happened. Our approach P^2F Miner makes use of frequent itemset mining to identify events and group sentences related to them. It uses an information-theoretic measure to rank identified events. For each of them, it selects a representative sentence as a description. Experiments on ClueWeb09 using events listed in Wikipedia year articles as ground truth show that our approach is effective and outperforms a baseline based on statistical language models.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Temporal Information Retrieval; Event Detection

1. INTRODUCTION

Good progress has been made during the last decade to semantically annotate documents and thus bring the goal of machines understanding natural language a bit closer. Examples of semantic annotations include named entities (e.g., persons, locations, and organizations), which have been the focus of named entity recognition and disambiguation [25, 29], and temporal expressions (e.g., during the 1990s) [15, 33]. It is nowadays possible to semantically annotate large-scale document collections consisting of billions of documents. Google, for example, has recently released [20] named entity annotations for the ClueWeb09/12 corpora which consist of about a billion documents each. Semantic annotations have been exploited to improve search – named entities in [19, 23]; temporal expressions in [10, 11]. Making use of semantic annotations to understand document contents at scale, however, is still in its infancy.

In this work, we adress one specific aspect of understanding document contents, namely identifying important events from the semantic annotations of a large-scale document collection. Given a time period of interest, specified as a temporal expression (e.g., 1879), our objective is to identify relevant events from that time period, rank them according to their importance, and present the user with a readable description of the event. The specified time period of interest may lie in the past, present, or future.

There are several challenges that we address along the way. First and foremost, the same real-world event can be expressed in natural language in countless ways. We thus need a way to group paraphrases, in our case sentences, that refer to the same event. To this end, our approach makes use of methods from frequent itemset mining to identify frequently co-occurring sets of named entities and groups sentences containing them. Second, the scale of document collections demands that methods be scalable. Our approach preprocesses, filters, and indexes the document collection as a one-time step, making retrieval of relevant sentences at query-processing time efficient. Third and last, we address a novel task with no existing benchmark that we could use for evaluation. Even worse, what constitutes an important event is highly subjective and views may differ based on origin or interests. As a remedy, we propose to and do use events as listed in Wikipedia year pages as ground truth. We argue that this is a reasonable approach, given Wikipedia's democratic and distributed nature. Different methods are then evaluated according to their ability to bring up ground-truth events listed in Wikipedia.

Contributions that we make in this work are:

- our *approach* P^2F Miner to identify important events from semantically-annotated document collections;
- an *experimental setup* that can be re-used to evaluate methods that address the same task;
- an *experimental evaluation* showing that our approach is effective and outperforms a baseline based on statistical language models.

Outline. The rest of this work is organized as follows. We lay out the technical foundation for our work in Section 2. Section 3 then describes our approach P^2F Miner in detail. Our experimental evaluation is subject to Section 4. We discuss related work in Section 5 and draw conclusions in Section 6.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2741692.

2. TECHNICAL BACKGROUND

We now lay out the technical foundation of our work.

Temporal Information. Documents come with different kinds of temporal information. Publication dates, if known accurately, reveal when the document was published. Temporal expressions can be extracted from document contents using a temporal expression tagger. State-of-the-art temporal taggers such as SUTime [15] rely on regular expressions defined on surface tokens and their part-of-speech (POS) tags. Alonso et al. [6] distinguish three kinds of temporal expressions. *Explicit* ones (e.g., February 2, 2015) are self-contained and can directly be resolved. *Implicit* temporal expressions (e.g., Christmas 2014) require background knowledge to be resolved. *Relative* temporal expressions (e.g., last month), finally, can only be resolved of the publication date of the document is known.

Named Entity Recognition & Disambiguation marks up mentions of named entities (e.g., persons, locations, organizations) in document contents. For instance, in the sentence "Google was founded by Stanford Ph.D. candidates Page and Brin", one would like to spot named entities such as Page and resolve them to a canonical named entity from a knowledge graph, for instance, `LarryPage` as opposed to `JimmyPage`. Knowledge graphs such as YAGO [24] and Freebase [13] contain millions of named entities, facts about them (e.g., `LarryPage wasBornIn EastLansing`), and assign them to semantic types (e.g., entrepreneur and scientist). Spotting of named entity mentions typically relies on learnt patterns – we refer to Sarawagi [29] for an overview. Disambiguating named entity mentions is more sophisticated and state-of-the-art approaches such as AIDA [25] are based on various collection statistics capturing the popularity of individual entities, their fit to the surrounding context, as well as coherence of entity pairs.

Event Detection. The Topic Detection and Tracking (TDT) initiative [16] defined an event as *"a particular thing that happens at a specific time and place, along with all necessary preconditions and unavoidable consequences."* As we will explain in the following section, our approach instruments this definition by looking at sentences that mention a specific time period together with a set of named entities.

Frequent Itemset Mining [35] has been proposed in the context of market basket analysis. Given a set of customer transactions, subsets of the universe of all items $\mathcal{I}$, the objective is to identify itemsets that occur in at least σ customer transactions. Put formally, for an itemset $X \subseteq \mathcal{I}$, we let $s(X)$ denote its support, that is, the number of customer transactions in which X occurs. Different algorithms [3, 21] have been proposed to mine all *frequent* itemsets whose support $s(X)$ is above the minimum support threshold σ. As a commonality, all algorithms exploit the anti-monotonicity of the itemset support (i.e., $X \subseteq X' \Rightarrow s(X) \geq s(X')$), which allows for a pruning of candidate itemsets. Thus, if the itemset $X = \{\mathsf{a}, \mathsf{b}\}$ is known to be infrequent, one can conclude that all its supersets including $X' = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ are infrequent. The number of frequent itemsets can be enormous, so that identifying and emitting all of them can become prohibitively expensive. Therefore, different ways to reduce the output have been proposed. One of them is to mine only *maximal* itemsets [14], from which the set of all frequent itemsets can be reconstructed. An itemset X is said to be *maximal* if no proper superset $X' \supsetneq X$ exists that is also frequent (i.e., $s(X') \geq \sigma$).

3. P²F MINER

Given a temporal expression of interest (e.g., 1879), our approach first retrieves relevant sentences from the document collection that mention a (related) temporal expression (e.g., March 14, 1897). In the next stage, these are analyzed and grouped to identify events mentioned therein. Following that, having distilled events, we rank discovered events according to their importance. Finally, for each event, a representative sentence is identified, which provides a meaningful description of the event. In the following, we provide details on these four stages of our approach P^{2}F Miner.

3.1 Sentence Retrieval

We assume that the document collection at hand, as a one-time preprocessing step, has been annotated with temporal expressions (e.g., March 1897) and disambiguated named entity mentions (e.g., `AlbertEinstein`). We keep all sentences that mention at least one temporal expression and one named entity. Given a temporal expression of interest to the user (e.g., 1879), as a first step, we need to retrieve all sentences that mention a relevant temporal expression.

We consider two variants of our approach which differ in which sentences they consider relevant to a specified temporal expression. The first variant, coined STRICT, retrieves only sentences that explicitly mention the specified temporal expression at a specific time granularity. For example, if the temporal expression 1879 is given, only sentences which mention 1879 at year granularity are retrieved and considered in the following stages. Our second variant, coined RELAXED, also considers more fine-grained temporal expressions which fall into the specified time period. Again, as a concrete example, given 1879 as an input, our second variant also considers sentences that mention a specific day (e.g., March 14, 1879) within the temporal expression of interest.

To facilitate efficient retrieval of relevant sentences, our approach relies on a temporal inverted index. Indexed terms correspond to temporal expressions (e.g., 19th Century) and for each of them a posting list is kept which records identifiers of sentences mentioning the temporal expression. As explained above, the second variant of our approach also considers more fine-grained temporal expressions (e.g., March 1879) when given a coarse-grained temporal expression of interest (e.g., 1879). Posting lists are laid out on disk to improve locality of access. More precisely, we ensure that posting lists belonging to adjacent temporal expressions of a specific granularity are stored contiguously. Thus, when given the temporal expression March 1879, so that posting lists for all days within that period need to be retrieved, we only have to read one contiguous block of data, avoiding expensive random accesses.

3.2 Event Distillation

Having identified all relevant sentences for the temporal expression of interest, we need to analyze their contents to identify important events mentioned therein. One challenge here is that the same event can be paraphrased in many different ways. We thus need a handle to group sentences that are likely to mention the same event.

To this end, we rely on the named entity annotations of relevant sentences. Following the event definition mentioned in Section 2, we group sentences based on the involved named entities (e.g., locations) that they mention. However, sentences may differ in the degree of detail in which they discuss an event. As a concrete example, Albert Einstein's birth may be reported by only indicating the place of birth `Ulm`, the state `BadenWürttemberg`, or also the country `Germany`. This suggests that we need to allow for small differences in the sets of named entities.

With this in mind, we identify maximal sets of named entities that are mentioned in relevant sentences. Using the terminology from Section 2, sentences correspond to customer transactions and named entities to items. Considering again our running example and assuming that the set of named entities $\{\texttt{AlbertEinstein}, \texttt{Ulm}, \texttt{Germany}\}$ is maximal, all sentences that mention a subset thereof are grouped together (e.g., $\{\texttt{AlbertEinstein}, \texttt{Ulm}\}$). This step hence yields a non-disjoint partitioning of the sentences and each partition is assumed to correspond to an event.

3.3 Event Ranking

Next, we need to rank the identified events based on their importance for the specified temporal expression of interest. For this step we rely on an established measure from information theory [18], namely Mutual Information (MI). MI measures the strength of association and correlation between two random variables and similar measures have been used in timeline generation [30]. In our setting the binary random variables T and I indicate whether a sentence mentions a temporal expression of interest (e.g., 1879) and a specific set of named entities (e.g., $\{\texttt{AlbertEinstein}, \texttt{Ulm}\}$). Mutual Information is then defined as

$$MI(T, I) = \sum_{t \in T} \sum_{i \in I} p(t, i) \log \left(\frac{p(t, i)}{p(t)p(i)} \right) .$$

Intuitively, a larger value for MI is observed when a set of named entities co-occurs with a temporal expression more often than would be expected by chance, which is what the denominator in the above equation captures. We determine MI for all identified events (maximal itemsets) and rank them in descending order of it.

3.4 Representative Selection

As a final step, before presenting the identified important events to the user, we need to come up with a meaningful description for each event. To this end, for each event, we select one of its belonging sentences as a representative. Which sentence is selected based on a set of heuristic rules based on the part-of-speech (POS) tags of the sentence. We thus require that the sentence must not start with a personal or possessive pronoun (e.g., *we* or *ours*) and that it contains a verb. If multiple sentences qualify, we select one randomly.

4. EXPERIMENTAL EVALUATION

In this section, we describe the experiments conducted to evaluate P^{2}F Miner and compare it against a baseline.

4.1 Dataset

We use ClueWeb09 [1] as a large-scale real-world document collection for our experiments. The dataset consists of about 1 billion web pages crawled in 2009. We concentrate on the subset of 503, 903, 810 web pages written in English. ClueWeb09 was processed using Stanford CoreNLP, that is, we determine sentence boundaries using the supplied standard model and annotate temporal expressions using the SUTime [15] component. For annotations of named entities, we make use of the recently released Google FACC annotations [20]. These are high-precision annotations that map named entity mentions to canonical named entities from the Freebase [13] knowledge graph. Sentences that mention at least one temporal expression, as discovered by SUTime, and contain at least one disambiguated named entity, as obtained from the Google FACC annotations, are retained and indexed as detailed in Section 3.1.

4.2 Methods under Comparison

Our experimental evaluation compares both variants of our approach, Strict and Relaxed, against a baseline based on statistical language models [36]. The baseline, coined Lm, uses the same set of relevant sentence as Strict when given a temporal expression T of interest. From the retrieved relevant sentences, the baseline estimates a unigram language model θ_T. For every sentence s, it estimates a unigram language model with Dirichlet smoothing θ_s. Relevant sentences are then ranked in ascending order of the Kullback-Leibler divergence between their language model θ_s and the language model θ_T. Intuitively, this brings up sentences related to important events whose constituting words have high probability in the language model θ_T.

4.3 Test Cases

The task that we address is relatively unexplored, so that there is no standard benchmark which we could use for our experimental evaluation. We therefore rely on Wikipedia as a ground-truth source of important events within a specific time period. More precisely, we make use of Wikipedia year articles (e.g., `http://en.wikipedia.org/wiki/1879`), extract the events listed therein, and assume that these are events of global importance that an effective method should identify. When looking at temporal granularities finer than a year (e.g., March 1879), we consider only the events listed in the corresponding section of the Wikipedia year article. Two caveats of this approach are that (i) some important events may not be covered in ClueWeb09, so that no method can identify them and (ii) Wikipedia year articles are incomplete, so that some identified events, although important, are not listed. We divide the timeline into four tenses, namely far past (1899 or earlier), past (1900 - 1999), present (2000 - 2009), and future (2010 and later). For each tense, we randomly pick a day, a month, and a year as temporal expressions of interest. This leaves us with a total of 36 test cases which we use for evaluation.

4.4 Relevance Assessments

To judge the effectiveness of the methods under comparison, we need to link retrieved sentences to events listed in Wikipedia year articles. To this end, we pool the sentences retrieved by the different methods and ask human assessor to link them to ground-truth events from the Wikipedia year article. This was implemented using Google Forms. Every test case is given to one human assessor. For every sentence retrieved by any of our methods a form is compiled that shows the sentence together with all ground-truth

March 14 1879	STRICT	*Albert Einstein was born on* 14th March, 1879 *in* `Ulm, Germany`
	RELAXED	*Albert Einstein was born on* 14th March, 1879 *in* `Ulm, Germany`
	LM	`Albert Einstein` *Physicist, 1879 - 1955* `Albert Einstein` *was born on* March 14, 1879 *in* `Ulm, Württemberg, Germany`
September 1870	STRICT	`The First Vatican Council` *ended when Italian soldiers entered* `Rome` *in* September 1870
	RELAXED	`Vatican city` *state celebrates 75 years on* February 11th, 1929, *an historic treaty was signed between the Italian Government and the* `Vatican` *re-establishing the political power and diplomatic standing of the* `Catholic Church`, *which had been lost when* `Italy` *seized* `Rome`, *the last of the Papal States, on* September 20th, 1870.
	LM	*For example,* `Engels` *asserted in his infamous diatribe "The Bakuninists at work" that Bakunin "[a]s early as* September 1870 *(in his Lettres a un francais [Letters to a France]) ...had declared that the only way to drive the Prussians out of* `France` *by a revolutionary struggle was to do away with all forms of centralised leadership and leave each town, each village, each parish to wage war on its own.*
1912	STRICT	*History:* `Montenegro` *joined* `Greece`, `Serbia` *and* `Bulgaria` *in a war against* `Turkey` *in 1912.*
	RELAXED	*The* `Treaty of Fez` *(signed on* March 30, 1912*) made* `Morocco` *a protectorate of* `France`.
	LM	*In* 1912 *he was elected to his first term as president of the* `US`.

Figure 1: Top-1 sentence retrieved by different methods for three of our test cases

events from Wikipedia retrieved for the temporal expression of interest. The assessor then selects zero, one, or multiple ground-truth events that the sentence relates to.

4.5 Effectiveness Measures

To measure the effectiveness of the different methods, we employ two families of effectiveness measures. First, we use Precision at rank k (P@k). One shortcoming of this is that it does not assess the novelty of retrieved sentences. Thus, a method could perform well under it, even if it only retrieves sentences related to a single ground-truth event from Wikipedia. As a second family of effectiveness measures, we therefore also use Intent-Aware Precision at rank k (IA-P@k), as proposed by Agrawal et al. [2], as well as α-DCG as proposed by Clarke et al. [17]. These measures reward novelty, or put differently, penalize redundancy in retrieved sentences. Query aspects (subtopics) for those measures are the ground-truth events from Wikipedia.

4.6 Anecdotal Results

We begin our discussion with some anecdotal results. Figure 1 shows the top-1 sentence retrieved by different methods for three of our test cases – a day, a month, and a year. While it is difficult to judge the historical significance of the events mentioned therein, the anecdotal results allow for the following observations. We can see that our methods STRICT and RELAXED agree on the most important event for two of the three test cases. The latter, by also looking at sentences that mention a more fine-grained temporal expression, broadens its scope and indeed, for two of the test cases, retrieves a sentence at the top which mentions a more fine-grained temporal expression (e.g., March 30, 1912 for the test case 1912). It also becomes clear that our methods profit from our heuristics for selecting a representative sentence – the baseline LM, in contrast, selects a hard-to-interpret sentence for the test case 1912, since the sentence does not mention the key entity `WoodrowWilson`.

4.7 Effectiveness Results

Table 1 shows the overall effectiveness results for our methods and the baseline. STRICT and RELAXED outperform the baseline markedly across both families of effectiveness measures. They thus do not only retrieve more relevant results, as indicated by P@k, but also do a better job at avoiding redundancy, as reflected by IA-P@k and α-DCG@k. Comparing our two methods, we observe that STRICT, retrieving only sentences that specifically mention the time period of interest, performs slightly better than RELAXED.

Method	Far Past P@10	Past P@10	Present P@10	Future P@10
STRICT	0.33	0.47	0.40	0.13
RELAXED	0.33	0.57	0.07	0.10
LM	0.13	0.07	0.07	0.20

Table 2: (P)recision@10 by tense.

To get a better understanding, we break down the results by tense (e.g., past) and granularity (e.g., year) – reported in Table 2 and Table 3. Interestingly, the baseline LM outperforms our methods when looking at time periods in the future. For all other tenses our methods are clearly ahead, but there is no clear winner between them. Looking at different granularities, we observe that all methods perform best (worst) for temporal expressions of interest at year (day) granularity. We speculate that this has to do with the prevalence of temporal expressions at year granularity in documents – a closer investigation is left for future work. STRICT and RELAXED again perform consistently better than the baseline.

Method	P@5	P@10	IA-P@5	IA-P@10	α-DCG@5	α-DCG@10
Strict	0.42	0.33	0.08	0.04	1.45	1.79
Relaxed	0.32	0.26	0.08	0.04	1.04	1.37
Lm	0.15	0.12	0.02	0.01	0.35	0.46

Table 1: (P)recision@(5, 10), IA-(P)recision@(5, 10), and α-DCG@(5, 10) ($\alpha = 0.5$)

Method	Day P@10	Month P@10	Year P@10
Strict	0.10	0.38	0.53
Relaxed	0.10	0.33	0.38
Lm	0.03	0.05	0.28

Table 3: (P)recision@10 by granularity.

Summary

Our methods Strict and Relaxed outperform the baseline based on statistical language models when considering all test cases derived from Wikipedia year articles. We observe that performance for all methods varies by tense and granularity. Test cases of finer granularity are harder – all methods perform worse for them.

5. RELATED WORK

We now put our work in context with existing prior research. This can be broadly categorized as follows:

Timelines. Automatically generating timelines from a collection of documents has been an active line of research for more than a decade. Swan et al. [30, 31] focus on news documents that come with a reliable publication date. Similar to our approach, they then rely on information-theoretic measures to extract features (named entities and noun phrases) that strongly correlate with a particular time period. More recently, timelines have been explored [5] in information retrieval as an alternative way to display search results. All of these methods rely on publication dates and do not exploit temporal expressions.

Topic Detection & Tracking. Allan [4] summarizes the outcomes of the topic detection and tracking (TDT) initiative, which addressed several tasks on streams of incoming (news) documents. The main focus here was on discovering topics (events) as they arise and track them while additional documents are published. Kuzey et al. [27], as a very recent work, looks into detecting named events from news sources and assigning a fine-grained semantic type to them (e.g., benefit_rock_concert); for evaluation, they also rely on events listed in Wikipedia. Our work differs from all of the above in looking at documents other than news and exploiting temporal expressions as opposed to only publication dates.

Temporal Information Retrieval. Making use of temporal information to improve information retrieval methods has seen significant interest in recent years. Baeza-Yates [10], as one of the earliest works, investigates how statements about the future can be retrieved and analyzed. The potential of temporal information has been described by Alonso et al. [6]. Concrete methods that make use of temporal expressions and/or publication dates include Arikan et al. [8], Berberich et al. [11], and Peetz et al. [28]. The focus in all of these, however, is on retrieving individual documents as opposed to analyzing a large collection of them. Indexing versioned document collections, such as web archives, has also received ample attention [7, 12, 22] – no existing work has looked into indexing temporal expressions contained in documents.

Temporal Information Extraction. Research in information extraction [32, 34] has looked into determining a temporal scope (or valid-time interval) of extracted facts. Typically, these methods are restricted to simple facts (i.e., binary relations) and can not easily deal with higher-order relations as required for real-world events.

Computational History. Yeung and Jatowt [9] and Jatowt and Yeung [26] is the work closest to ours. The focus in the former is on studying how the past is remembered; in the latter the focus is on analyzing expectations about the future. Both approaches rely on topic modeling to group documents that relate to similar events in the past or future. Typical topics are more coarse-grained than the events that we target. Moreover, topics are described by word distributions, which loose word order and are thus less readable than our representative sentences.

6. CONCLUSION

We have put forward a novel approach to identify important events, that happened during a time period of interest, from a semantically-annotated document collection. Two variants of our approach were described and we compare them against a baseline based on statistical language models. Our experiments on ClueWeb09 using lists of events from Wikipedia year articles as ground truth show that our approach is effective and outperforms the baseline.

As part of our ongoing research, we investigate how we can also make use of geographic references with their inherent semantics (e.g., `Munich` lies in `Bavaria` lies in `Germany`). Further, we are working on improving the efficiency of the system to make response times truly interactive.

7. REFERENCES

[1] ClueWeb09 `http://lemurproject.org/clueweb09/`.

[2] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. *WSDM* 2009

[3] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. *SIGMOD* 1993

[4] J. Allan. Introduction to topic detection and tracking. *Topic Detection and Tracking*, Springer, 2002

[5] O. Alonso, K. Berberich, S. Bedathur, and G. Weikum. Time-based exploration of news archives. *HCIR* 2010
[6] O. Alonso, M. Gertz, and R. Baeza-Yates. On the value of temporal information in information retrieval. *SIGIR Forum* 2007
[7] A. Anand, S. J. Bedathur, K. Berberich, and R. Schenkel. Index maintenance for time-travel text search. *SIGIR* 2012
[8] I. Arikan, S. J. Bedathur, and K. Berberich. Time will tell: Leveraging temporal expressions in IR. *WSDM* 2009
[9] C.-m. Au Yeung and A. Jatowt. Studying how the past is remembered: towards computational history through large scale text mining. *CIKM* 2011
[10] R. Baeza-Yates. Searching the Future. *MF/IR* 2005
[11] K. Berberich, S. Bedathur, O. Alonso, and G. Weikum. A language modeling approach for temporal information needs. *ECIR* 2010
[12] K. Berberich, S. J. Bedathur, and G. Weikum. Efficient time-travel on versioned text collections. *BTW* 2007
[13] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. *SIGMOD* 2008
[14] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu. Mafia: A maximal frequent itemset algorithm. In *TKDE* 17:1490–1504, 2005
[15] A. X. Chang and C. D. Manning. Sutime: A library for recognizing and normalizing time expressions. *LREC* 2012
[16] C. Cieri, S. Strassel, D. Graff, N. Martey, K. Rennert, and M. Liberman. Corpora for topic detection and tracking. *Topic Detection and Tracking*, Springer, 2002
[17] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. *SIGIR* 2008
[18] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
[19] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. *SIGIR* 2014
[20] E. Gabrilovich, M. Ringgaard, and A. Subramanya. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0), 2013.
[21] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *SIGMOD* 2000
[22] J. He, J. Zeng, and T. Suel. Improved index compression techniques for versioned document collections. *CIKM* 2010
[23] J. Hoffart, D. Milchevski, and G. Weikum. STICS: searching with strings, things, and cats. *SIGIR* 2014
[24] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. In *Artif. Intell.*, 194:28–61, 2013
[25] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. *EMNLP* 2011
[26] A. Jatowt and C.-m. Au Yeung. Extracting collective expectations about the future from large text collections. *CIKM* 2011
[27] E. Kuzey, J. Vreeken, and G. Weikum. A fresh look on knowledge bases: Distilling named events from news. *CIKM* 2014
[28] M.-H. Peetz, E. Meij, and M. de Rijke. Using temporal bursts for query modeling. In *Inf. Retr.*, 17(1):74–108, 2014
[29] S. Sarawagi. Information extraction. In *Found. Trends Databases*, 1(3):261–377, 2008.
[30] R. Swan and J. Allan. Automatic generation of overview timelines. *SIGIR* 2000
[31] R. Swan and D. Jensen. Timemines: Constructing timelines with statistical models of word usage. *KDD Workshop on Text Mining* 2000
[32] P. P. Talukdar, D. T. Wijaya, and T. M. Mitchell. Coupled temporal scoping of relational facts. *WSDM* 2012
[33] M. Verhagen, I. Mani, R. Sauri, J. Littman, R. Knippen, S. B. Jang, A. Rumshisky, J. Phillips, and J. Pustejovsky. Automating temporal annotation with tarsqi. *ACL* 2005
[34] Y. Wang, M. Dylla, M. Spaniol, and G. Weikum. Coupling label propagation and constraints for temporal fact extraction. *ACL* 2012
[35] M. J. Zaki and J. Wagner Meira. *Data Mining and Analysis: Fundamental Concepts and Algorithms.* Cambridge University Press, 2014.
[36] C. Zhai. Statistical language models for information retrieval a critical review. In *Found. Trends Inf. Retr.*, 2:137–213, 2008.