# Bridging Temporal Context Gaps using Time-Aware Re-Contextualization

Andrea Ceroni, Nam Khanh Tran, Nattiya Kanhabua, and Claudia Niederée

L3S Research Center / Leibniz Universität Hannover, Germany

{ceroni, ntran, kanhabua, niederee}@L3S.de

## ABSTRACT

Understanding a text, which was written some time ago, can be compared to translating a text from another language. Complete interpretation requires a mapping, in this case, a kind of time-travel translation between present context knowledge and context knowledge at time of text creation. In this paper, we study *time-aware re-contextualization*, the challenging problem of retrieving concise and complementing information in order to bridge this temporal context gap. We propose an approach based on learning to rank techniques using sentence-level context information extracted from Wikipedia. The employed ranking combines relevance, complementarity and time-awareness. The effectiveness of the approach is evaluated by contextualizing articles from a news archive collection using more than 7,000 manually judged relevance pairs. To this end, we show that our approach is able to retrieve a significant number of relevant context information for a given news article.

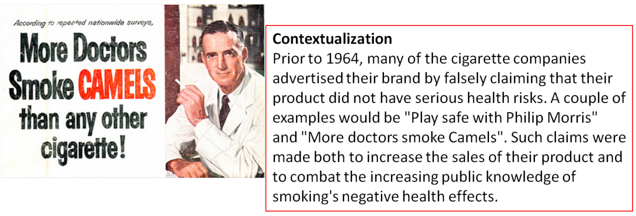
**Categories and Subject Descriptors** H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

**General Terms** Algorithms, Experimentation

**Keywords** Time-aware re-contextualization, Temporal context, Complementarity, Wikipedia

## 1. INTRODUCTION

Reading a current news article about your own country typically is straightforward. Things get worse if the article is, for example, from the 60s or the 70s as it can be found in news archives such as the New York Times Archive[1]. We are especially interested in time-aware *re-contextualization* settings, where explicit context information is required for bridging the gap between the situation at the time of content creation and the situation at the time of content digestion. This includes changes in background knowledge, the societal and political situation, language, technology, and simply the forgetting of the original knowledge about the context.



**Contextualization**
Prior to 1964, many of the cigarette companies advertised their brand by falsely claiming that their product did not have serious health risks. A couple of examples would be "Play safe with Philip Morris" and "More doctors smoke Camels". Such claims were made both to increase the sales of their product and to combat the increasing public knowledge of smoking's negative health effects.

**Figure 1: Camel advertisement (left) and contextualization information taken from Wikipedia (right).**

The importance of time-aware re-contextualization is well illustrated by the advertisement poster from the 1950s in Figure 1. From today's perspective it is more than surprising that it would be actually doctors, who recommend smoking. It can, however, be understood from the context information at the right side of Figure 1, which has been extracted from the Wikipedia article on tobacco advertising.

Dealing with content from former times is not restricted to expert users such as journalists, historians or researchers. With the growing age of the Web, general Web users are increasingly confronted with content, which has been created at different time points in the past, assuming knowledge of the context at the respective time for its interpretation.

Basic forms of contextualization have already been suggested in early works such as [4, 10, 11]. The Wikify! system [10], for example, enables an automated linkage of concept mentions with Wikipedia pages. Pure linkage to the Wikipedia article is, however, not sufficient for the re-contextualization task we are targeting. First, Wikipedia pages on popular concepts, events and entities tend to contain large amounts of content, while the concrete aspect of the text to be contextualized might be covered only marginally or not at all; and relevant information might be distributed over various articles. Furthermore, the crucial temporal aspect such as considering the situation with respect to smoking in the 1950s is also missing in pure linking approaches.

Time-aware re-contextualization, that is, the association of an information item $i$ (such as a phrase in a text) with additional context information $c_i$ for easing its understanding is a challenging task. Several subgoals of the information search process have to be combined with each other: (1) $c_i$ has to be relevant for $i$, (2) $c_i$ has to complement the information already available in $i$ and the surrounding document, (3) $c_i$ has to consider the time of creation (or reference) of $i$, and (4) the set of collected context information for $i$ should be concise avoiding to overload the user.

[1] http://catalog.ldc.upenn.edu/LDC2008T19

*SIGIR'14,* July 6–11, 2014, Gold Coast, Queensland, Australia.
Copyright 2014 ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609526.

Our main contributions are: (1) framing the problem of time-aware re-contextualization; (2) a learning-to-rank approach combining temporal aspects with the idea of complementarity; and (3) its evaluation with over 7,000 relevance judgments using two real-world document collections.

## 2. RELATED WORK

As discussed in Section 1, our work goes beyond the pure linkage of entity and concept mentions with Wikipedia pages as it is described in [4, 10, 11].

In the area of temporal search, previous work [2] has shown that leveraging the time dimension in ranking can improve the retrieval effectiveness for temporal queries. Retrieving and processing external information to be added to documents has been widely studied in the recent years. In [6], for example, news articles are enriched with related predictions retrieved from other documents in the same collection. In [3], the authors present a topic modeling approach which jointly exploits news articles and Twitter for event summarization. In order to generate a representative but not redundant summary of an event, complementarity between tweets and news article sentences is assessed by considering both their similarity and their difference. In contrast to those approaches, our work on time-aware re-contextualization adds another dimension to the contextualization task, namely time. We are not looking for more information on the current context, but try to re-construct the original context of a document.

From an application perspective, our work is also related to computational history, which refers to the application of data analysis and mining techniques in support of history research. In this field, various methods have been developed including methods for speeding up search and analyses [5], methods for investigating linguistic and cultural trends [9], as well as methods for analyzing collective memory and the perception based on news article references to the past [1].

## 3. PROBLEM DEFINITION

Given a document $d$ with creation date $t_d$ and a source of background information $C$ (or a *context source*), we define *time-aware re-contextualization* as the process of reconstructing the relevant part of the original context of document $d$ at time $t_d$ by retrieving information from $C$ that helps in interpreting $d$.

In more detail, time-aware re-contextualization can be regarded as a combination of an annotation problem with a retrieval and ranking problem. The annotation problem consists of identifying a set of (possibly related) *contextualization hooks* together with temporal references. These are the parts of $d$ that require contextualization. Contextualization hooks can, for example, be entity mentions, concept mentions, implicit topics, or phrases. The retrieval problem consists of identifying context units within a context source $C$ that are candidates for contextualizing context hooks. $C$ can for example be an ontology composed from statements about instances as context units or a document collection composed from sentences as context units. Finally, there is a need for ranking contextualization candidates in a way that top-k results of the ranking form a concise, useful and diverse set of contextualization units.

## 4. OUR APPROACH

We use a textual collection of background knowledge as context source, and *augmented sentences* as context units. An augmented sentence $c$ is a tuple $c := (s_c, T_c, E_c, p_c)$, where $s_c$ is the text of the sentence augmented with the text of its previous and following sentences, $T_c$ is a set of temporal expressions present in $s_c$, $E_c$ is a set of entities mentioned in $s_c$, and $p_c$ is the title of the document the sentence belongs to. We augment sentences with their neighbors under the assumption that a single sentence usually does not contain sufficient information for being understood in isolation. In the rest of this paper, we will use the terms *augmented sentences* and *sentences* interchangeably.

Given a document $d$ to be contextualized, we construct a set of queries from its textual content, and retrieve sentences (or context units) using the constructed queries, and re-rank them using a ranking model. Particularly, we address the ranking problem by defining and investigating different features for achieving effective time-aware re-contextualization. For this reason, we introduced some simplifications that allowed us to easily formulate queries and retrieve sentences to re-rank, leaving the investigation of how to automatically identify what requires additional information as a future work. Next, we will describe the query formulation and present the features used for learning a ranking model.

### 4.1 Query Formulation

To retrieve context units for a given document $d$, a set of queries has to be formulated capturing the parts of $d$ that require re-contextualization for being understood. Identifying such parts is not a trivial task, since they might depend on the user's background knowledge as well as on the publication date of the document. To gain insights, we conducted a preliminary study with a group of human evaluators asking them to annotate the parts of a set of older news articles that require additional information. Over a total of 221 annotations, 37% represented entities, 32% concepts and topics, 16% terms, and 15% short phrases. This means that employing entity and topic extraction tool alone is not sufficient, especially because not all the detected entities and topics would require re-contextualization. Even assuming to use these tools, there is still the problem of how to combine different hooks, that are logically related, in a single query.

For these reasons, we decided to simplify the query formulation step by asking evaluators to build queries for a given document: given a document $d$, a set $Q_d$ of queries is built by manually selecting and combining words within the document. The publication date $t_d$ is not included in the query, since this might prevent the system to retrieve useful sentences whose temporal expression set $T_c$ is empty. The temporal dimension will be exploited in the re-ranking model (Section 4.2). For each query $q$ within the set $Q_d$ the top-$k$ sentences are retrieved from the context source. The retrieved results are stored in different ranked list $C_{d,q}$.

### 4.2 Ranking

We propose different features for ranking a set of contextualization candidates $C_{d,q}$, including 4 classes of features that are used to estimate the usefulness of a sentence in reconstructing the original context of a document $d$.

**Temporal Similarity.** The first class of feature is aimed at capturing temporal similarity, or measuring how close the temporal expressions in a sentence $c$ are to the creation time $t_d$ of a document $d$. The intuition behind this is that if $c$ contains temporal expressions that are close to $t_d$, it is more likely to contain information referring to the situation at time $t_d$. In order to do that, we employ a time-decay function TSU [6, 7], which is computed as:

$$TSU(t_1, t_2) = \alpha^{\lambda^{\frac{|t_1 - t_2|}{\mu}}} \tag{1}$$

where $\alpha$ and $\lambda$ are constants, $0 < \alpha < 1$ and $\lambda > 0$, and $\mu$ is a unit of time distance. The value of this function decreases exponentially with respect to the time distance between $t_1$ and $t_2$. Given a sentence $c$ and a document $d$, we compute the maximum and the average temporal similarities between them as: $TSU_{max} = \max_{t \in T_c}\{TSU(t, t_d)\}$ and $TSU_{avg} = \frac{1}{\|T_c\|}\sum_{t \in T_c}\{TSU(t, t_d)\}$.

**Term Similarity.** Using only a temporal similarity is not sufficient for re-contextualizing documents because not all information temporally close to the creation date of the document is relevant to it. For this reason, we include the original *tf-idf* score assigned by the search engine to each retrieved sentence as a learning feature.

**Complementarity.** In some cases, a sentence can be related to a document without adding any useful information for understanding, e.g. the information might be already present in the document. Thus, we consider the complementarity of sentences to the document, which has been introduced in [3] for summarization tasks. Given two texts $s_1$ and $s_2$, a complementarity metric *compl* is computed as:

$$compl(s_1, s_2) = \begin{cases} \frac{sim(s_1, s_2)}{dif(s_1, s_2)} & \text{if } sim(s_1, s_2) \leq dif(s_1, s_2) \\ \frac{dif(s_1, s_2)}{sim(s_1, s_2)} & \text{otherwise} \end{cases} \tag{2}$$

where $sim(s_1, s_2)$ and $dif(s_1, s_2)$ are the similarity and difference between $s_1$ and $s_2$ respectively. We compute text-based and entity-based complementarities between a sentence $c$ and a document $d$ as: $compl_t(s_c, s_d)$ and $compl_e(E_c, s_d)$. We use Jaccard Index to compute similarity and difference between two amounts of text $s_1$ and $s_2$. Before computing these measures, text and entities have been preprocessed with stop-words removal, tokenization and stemming.

**Sentence-Based Features.** We consider 3 additional features based on information present in sentences. First, we measure how much the title field $p_c$ of a sentence is mentioned in a document as $title = \frac{matches}{\|p_c\|}$, where *matches* is the number of words of the title that are present in the document, and $\|p_c\|$ is the number of words that form the title. Second, we consider the *length* of sentences as feature. Finally, we compute the number of entity mentions within a sentence as $e_{\%} = \frac{\|E_c\|}{\|s_c\|}$.

**Ranking Model.** In order to jointly consider the proposed features in a unique ranking model, we resort to the *learning to rank* paradigm [8]. In our experiments, we employ different learning to rank algorithms, namely AdaRank, RankBoost, RankNet, ListNet, and LambdaMART. However, the best performing algorithms are AdaRank and RankBoost. For this reason, we will report and discuss the results only for these two ranking algorithms.

# 5. EXPERIMENTS

In this section, we describe our experimental settings, present the results followed by a detailed discussion.

## 5.1 Experimental Settings

**Document Collections.** In our experiments, we used the New York Times Annotated Corpus, which contains 1.8 million documents from January 1987 to June 2007, as the document collection to be re-contextualized. We employed the Wikipedia dump of February 2013 as a context source, and used Stanford CoreNLP parser[2] for tokenization, sentence splitting, entity annotation, and temporal expression extraction. After splitting Wikipedia pages in sentences, the text of every sentence has been augmented with the text of its previous and following sentences, according to Section 4; entities and temporal expressions have been then extracted from the augmented text. We used Apache Solr[3] to index the augmented sentences, obtaining more than 70 millions of indexed sentences. Entities, temporal expressions, and the title of the Wikipedia page containing the augmented sentence where stored in separate fields.

**Query Documents.** In our evaluation, we manually selected 30 news articles to be contextualized from different topics (politics, science, education, sport, and wars). In particular, we chose news articles from the earlier years of the collection because of their higher need for time-aware re-contextualization. For each article, we then constructed a set of queries, as described in Section 4.1, to retrieve relevant sentences from the Wikipedia index. We exploited only the lead paragraph of articles because it provides a concise summary of the most important topics in the article.

**Relevance Assessment.** Since there is no gold standard for the time-aware re-contextualization task, we built a ground truth by retrieving top-$k$ sentences for each formulated query by using Solr default similarity scoring function with $k = 100$. In more detail, we asked human assessors to evaluate query/sentence pairs using 4 levels of relevance: 3 for *excellent* (very relevant context), 2 for *good* (relevant context), 1 for *fair* (related context), 0 for *bad* (non-relevant context). The human relevance assessment took into account the requirement for providing *additional information* which complements the information in the lead paragraph of a given query article. More precisely, a retrieved sentence is relevant context, if it is relevant and if it provides contextual information not already present in the article. Finally, we considered a pair ($q$,$s$) as *relevant* if its relevance score is greater than 1, otherwise it is regarded as *irrelevant.* In total, we evaluated 7,390 query/sentence pairs[4], which are used for training the learning to rank algorithms.

**Parameter Settings.** We used the learning-to-rank implementation of RankLib[5]. For AdaRank, we set a training iteration to 100, the tolerance between consecutive learning rounds to $2 \cdot 10^{-3}$, and the maximum number of consecutive feature selection to 3. For RankBoost, we set the number of rounds to 100, and the number of threshold candidates to 5. We chose these parameter settings because they achieved the best performance. The ranking models were trained via 5-fold cross validation. For TSU, we set $\lambda = 0.25$, $\alpha = 0.5$, and $\mu = 2y$, where $y$ is the number of years.

## 5.2 Results

We evaluated our approach for re-contextualization compared to Solr default ranking (*tf-idf*), which is considered as baseline. Different ranking models have been trained by combining the features described in Section 4 in different

[2] `http://nlp.stanford.edu/software/corenlp.shtml`

[3] `https://lucene.apache.org/solr/`

[4] The evaluation collection we have created is available for download at: `http://www.l3s.de/~ceroni/SIGIR_2014/dataset.rar`

[5] `http://sourceforge.net/p/lemur/wiki/RankLib/`

ways, showing how the individual features contribute to the task. We report the results for both learning to rank algorithm employed.

In our task, we focus on top-precision performance metrics instead of recall-based metrics: we assume that users are interested in few very useful re-contextualizing sentences, and that a great number of false positives would annoy the user during the reading. Thus, we will measure the ranking performances through the precision at 1, 3, 5, and 10 (P@1, P@3, P@5, and P@10), as well as Mean Average Precision (MAP). The reported performances are average values over the 5 folds that we created to train the models.

In Table 1, we report the results obtained by using different feature sets in the learning model. The symbol * indicates statistically improvement over the baseline using t-test with significant at $p < 0.05$. The model considering the whole set of features (*all* in the table) achieves the best performances for each of the evaluation criteria and algorithms. It reaches precision values from 0.601 (P@10, RankBoost) to 0.774 (P@1, RankBoost), with improvements between 28.7% and 45.5% over the baseline. Also *TSU* and *compl*, i.e., the models considering temporal and complementarity features, respectively, outperform $tf - idf$. *TSU* performs better than *compl* under most of the evaluation criteria, confirming that the temporal dimension alone gives significant insights in estimating sentence relevance. For the sake of completeness, we also report results obtained with RankBoost when considering individual features separately in the model. These are complementarity based on entities ($compl_e$), complementarity based on text ($compl_t$), maximal temporal similarity $TSU_{max}$ and average temporal similarity $TSU_{avg}$. We do not report detailed behaviors for each sentence-based feature within $sent - based$ because they did not provide relevant results when considered alone.

In order to further investigate the impact of the individual features for the task of time-aware re-contextualization, we also evaluated the effect of excluding -in turn- each class of features (temporal similarity, complementarity, tf-idf, sentence-based) from the training process. The results are reported in Table 2. For both learning algorithms, we can clearly observe that the greatest decrease of performances occurs when the temporal features are excluded (*no_TSU*). A performance decrease can be also noticed for the *no_compl* model, but it is less pronounced and in some cases it is comparable with the decrease of $no\_tf - idf$. This behavior suggests that the temporal aspect has a higher influence than the complementarity for deciding if a sentence is relevant for re-contextualization. However, further investigations and experiments are envisioned to assess the role of complementarity in time-aware re-contextualization. In facts, our complementarity measure is based on assessing similarity and difference between text via a term-based metric (Jaccard Index), which cannot capture similarity and differences between content at a semantic level. Therefore, we plan to experiment with semantic approaches (e.g. topic modeling) for better assessing complementarity between two texts.

| | **Features** | **P@1** | **P@3** | **P@5** | **P@10** | **MAP** |
|---|---|---|---|---|---|---|
| | *tf-idf* | 0.547 | 0.491 | 0.481 | 0.469 | 0.451 |
| RankBoost | *all* | 0.774* | 0.711* | 0.664* | 0.601* | 0.622* |
| | *compl* | 0.696* | 0.583* | 0.557* | 0.499 | 0.515* |
| | *TSU* | 0.768* | 0.677* | 0.620* | 0.556* | 0.567* |
| | *sent-based* | 0.545 | 0.527 | 0.515 | 0.478 | 0.487* |
| | $compl_e$ | 0.649 | 0.575* | 0.537* | 0.491 | 0.490* |
| | $compl_t$ | 0.571 | 0.515 | 0.516 | 0.522* | 0.490* |
| | $TSU_{max}$ | 0.758* | 0.684* | 0.653* | 0.564* | 0.572* |
| | $TSU_{avg}$ | 0.753* | 0.640* | 0.569* | 0.537* | 0.546* |
| AdaRank | *all* | 0.642* | 0.650* | 0.658* | 0.598* | 0.553* |
| | *compl* | 0.471 | 0.585* | 0.522 | 0.511 | 0.512* |
| | *TSU* | 0.616 | 0.570 | 0.607* | 0.546* | 0.541* |
| | *sent-based* | 0.374 | 0.427* | 0.448 | 0.441 | 0.459 |

**Table 1: Overall performance of different features.**

| | **Features** | **P@1** | **P@3** | **P@5** | **P@10** | **MAP** |
|---|---|---|---|---|---|---|
| RankBoost | *all* | 0.774* | 0.711* | 0.664* | 0.601* | 0.622* |
| | *no_tf-idf* | 0.774* | 0.690* | 0.654* | 0.591 | 0.614* |
| | *no_compl* | 0.753* | 0.697* | 0.660* | 0.597* | 0.607* |
| | *no_TSU* | 0.579 | 0.559 | 0.567* | 0.535* | 0.538* |
| | *no_sent-based* | 0.826* | 0.706* | 0.670* | 0.610 | 0.609 |
| AdaRank | *all* | 0.642* | 0.650* | 0.658* | 0.598* | 0.553* |
| | *no_tf-idf* | 0.642* | 0.654* | 0.671* | 0.598* | 0.552* |
| | *no_compl* | 0.696* | 0.664* | 0.693* | 0.570* | 0.434 |
| | *no_TSU* | 0.720* | 0.598* | 0.547* | 0.521* | 0.519* |
| | *no_sent-based* | 0.629* | 0.716* | 0.679* | 0.618* | 0.555* |

**Table 2: Impact of removing features from learning.**

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an approach for time-aware re-contextualization of texts, in our case news articles, using Wikipedia as the contextualization source. The experiments showed that contextualization sentences can be identified with considerable precision, when combining temporal features with the idea of complementarity in re-ranking relevant search results for the purpose of re-contextualization.

As future work, we plan to further develop the approach for time-aware re-contextualization including the consideration of the semantic level in computing complementarity and the development of approaches for automatically identifying context hooks and the queries deduced from them.

**Acknowledgments** The work was partially funded by the European Commission for the FP7 project ForgetIT under grant No. 600826 and the ERC Advanced Grant ALEXANDRIA under grant No. 339233, and funded by the German Federal Ministry of Education and Research (BMBF) for the project *"Gute Arbeit" nach dem Boom (Re-SozIT)* (01UG1249C).

## 7. REFERENCES

[1] C.-m. Au Yeung and A. Jatowt. Studying how the past is remembered: towards computational history through large scale text mining. In *CIKM*, 2011.

[2] K. Berberich, S. J. Bedathur, O. Alonso, and G. Weikum. A language modeling approach for temporal information needs. In *ECIR*, 2010.

[3] W. Gao, P. Li, and K. Darwish. Joint topic modeling for event summarization across news and social media streams. In *CIKM*, 2012.

[4] 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. In *EMNLP*, 2011.

[5] L. Hoffmann. Looking back at big data. *Commun. ACM*, 2013.

[6] N. Kanhabua, R. Blanco, and M. Matthews. Ranking related news predictions. In *SIGIR*, 2011.

[7] N. Kanhabua and K. Nørvåg. Determining time of queries for re-ranking search results. In *ECDL*, 2010.

[8] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3), 2009.

[9] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, T. G. B. Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 2011.

[10] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM*, 2007.

[11] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, 2008.