Extracting News Web Page Creation Time with DCTFinder

Xavier Tannier

LIMSI-CNRS
Univ. Paris-Sud, 91403 Orsay, France
xavier.tannier@limsi.fr

Abstract

Web pages do not offer reliable metadata concerning their creation date and time. However, getting the document creation time is a necessary step for allowing to apply temporal normalization systems to web pages. In this paper, we present DCTFinder, a system that parses a web page and extracts from its content the title and the creation date of this web page. DCTFinder combines heuristic title detection, supervised learning with Conditional Random Fields (CRFs) for document date extraction, and rule-based creation time recognition. Using such a system allows further deep and efficient temporal analysis of web pages. Evaluation on three corpora of English and French web pages indicates that the tool can extract document creation times with reasonably high accuracy (between 87 and 92%).

DCTFinder is made freely available on `http://sourceforge.net/projects/dctfinder/`, as well as all resources (vocabulary and annotated documents) built for training and evaluating the system in English and French, and the English trained model itself.

Keywords: temporal analysis, web, CRF

1. Introduction

In most applications performing textual temporal analysis, addressed texts are newswire articles from only few sources. Considered corpora have a well-defined structure, with associated metadata – authors, title and document creation time (DCT). Only a few systems have been run on open-domain web pages, for two main reasons:

- Web pages need to be cleaned before a proper analysis is performed on the text. The main textual content of a web page must be extracted, and menus, ads and non-informative content must be filtered out.
- As long as HTML5 `pubdate` tag does not become widespread, there is no reliable metadata providing the web page creation time. However, document creation time is a necessary clue to analyze and normalize correctly most of the temporal expressions (Llorens et al., 2012; Strötgen and Gertz, 2012; Chang and Manning, 2012). These temporal expressions are often relative (*e.g. "Tuesday" or "June 21st"*), and their reference time is often the DCT.

The first issue is addressed by Web page cleaners such as BodyTextExtraction (Finn et al., 2001), Boilerpipe (Kohlschtter et al., 2010), jusText (Pomikálek, 2011) or Readability[1]. Concerning the second one, systems running on web pages only analyze absolute dates and are not able to extract any relative temporal information. However, Kessler et al. (2012) states that only 7% of the dates in news are absolute dates.

1.1. Motivation

Knowing the web page creation date would allow to apply temporal normalization systems to web pages, and then to open the way to much more knowledge and application fields. However, even if almost all news web pages are time-stamped, there is no straightforward way to get their creation date: no metadata or server information are reliable, and all sites have a different way to insert the date in the HTML content. A lot of dates occur in a web page, but only one is the creation date (see an example in Figure 1).

Many applications using temporal analysis or temporal knowledge could benefit from a system extracting title and document creation time from a web page, so that much more temporal information can coincide with general knowledge from the Web.

By combining such a system to a web page cleaner, we could analyze web pages with the same information as provided by a well-structured collection of newswire articles, *i.e.* a title, a DCT and an informative textual content.

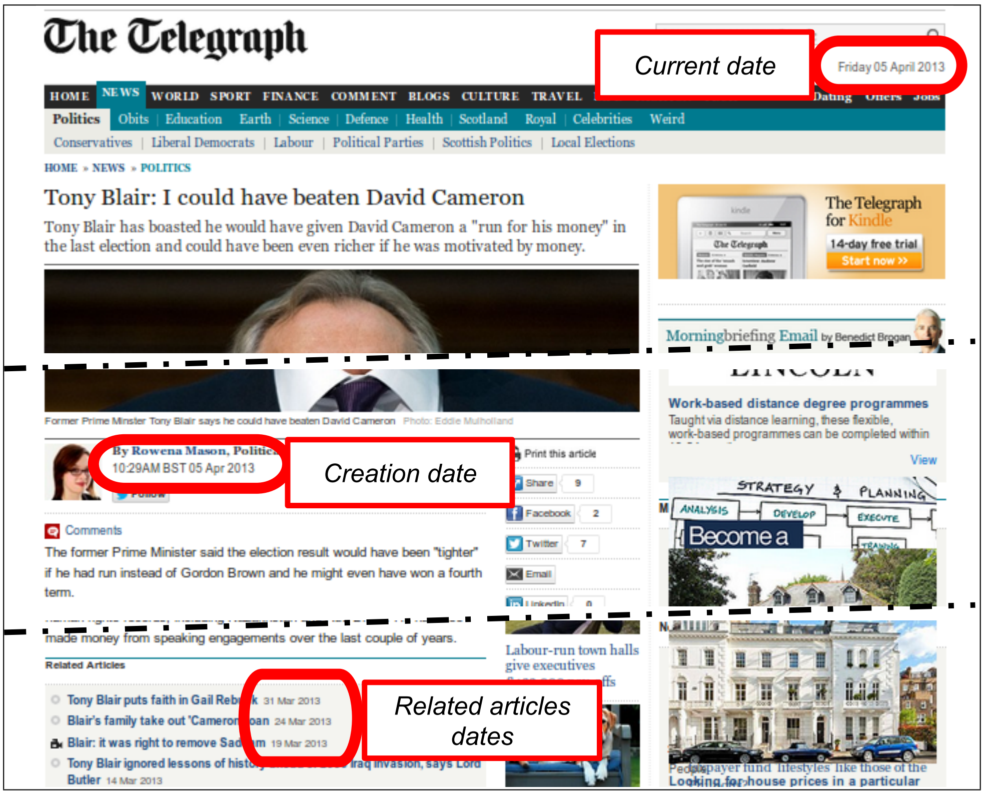
1.2. Related Work

Temporal analysis of texts is a very dynamic research field in natural language processing (NLP). It has received growing attention in the 2000s (Mani et al., 2005) and has lead to some standardization effort through the TimeML initiative (Pustejovsky et al., 2010).

Alonso et al. (2007), Alonso (2008), Kanhabua (2009), among others, have highlighted that the analysis of temporal information is often an essential component in text understanding and is useful in a wide range of NLP and information retrieval applications.

Harabagiu and Bejan (2005) and Saquete et al. (2009) highlight the importance of processing temporal expressions in Question Answering systems. Temporal processing enables a system to detect redundant excerpts from various texts on the same topic and to present results in a relevant chronological order, wich is crucial in multi-document summarization (Barzilay and Elhadad, 2002) or for building automatic timelines from a query (Kessler et al., 2012), as

[1] `http://www.readability.com/`



The Telegraph
Current date
Friday 05 April 2013
HOME NEWS WORLD SPORT FINANCE COMMENT BLOGS CULTURE TRAVEL Dating Offers Jobs
Politics Obits Education Earth Science Defence Health Scotland Royal Celebrities Weird
Conservatives | Liberal Democrats | Labour | Political Parties | Scottish Politics | Local Elections
HOME » NEWS » POLITICS
The Telegraph for Kindle
14-day free trial
Start now »
Morningbriefing Email by Benedict Brogan
Creation date
Print this article
Share 9
Facebook 2
Twitter 7
Email
LinkedIn 0
Comments
View
Related Articles
Related articles dates

Tony Blair: I could have beaten David Cameron

Tony Blair has boasted he would have given David Cameron a "run for his money" in the last election and could have been even richer if he was motivated by money.

Former Prime Minster Tony Blair says he could have beaten David Cameron Photo: Eddie Mulholland

By Rowena Mason, Politic
10:29AM BST 05 Apr 2013

The former Prime Minister said the election result would have been "tighter" if he had run instead of Gordon Brown and he might even have won a fourth term.

made money from speaking engagements over the last couple of years.

Work-based distance degree programmes
Taught via distance learning, these flexible, work-based programmes can be completed within

Labour-run town halls give executives

- Tony Blair puts faith in Gail Reb k 31 Mar 2013
- Blair's family take out 'Cameron oan 24 Mar 2013
- Blair: it was right to remove Sad m 19 Mar 2013
- Tony Blair ignored lessons of hist Iraq invasion, says Lord Butler 14 Mar 2013

Figure 1: A typical news web page with different dates.

well as for aiding medical decision-making (Kim and Choi, 2011). Similarly, Jung et al. (2011) present an end-to-end system that processes clinical records, detects events and constructs timelines of patients' medical histories. Finally, the evaluation campaigns TempEval 2007 (Verhagen et al., 2007) and TempEval 2010 (Verhagen et al., 2010) (and currently running TempEval 2013) focused on temporal relation identification within a single text.

For all these applications, normalizing temporal expressions is necessary. Systems like Heideltime (Strötgen and Gertz, 2013), SUTime (Chang and Manning, 2012), Timen (Llorens et al., 2012) or ManTIME (Filannino et al., 2013) are dedicated to this task, which can only be performed if the time of creation of the parsed document is known.

2. Extracting Web Page Creation Time with CRFs

Many dates occur in a web page, but only one is the document creation time (DCT). Some pages include also the current date ("*now*") or the date of last update. A lot of pages also provide links to other articles from the same site, with their associated dates. Finally, dates are present in the textual content of the article. For these reasons, the challenge is not to detect dates, which is quite simple, but rather to select the DCT from all detected dates.

One important clue for achieving this selection is the proximity to the page title. Unfortunately, finding the page title is not that easy that we could think. For efficiency reasons, we do not want to rely on page rendering techniques. HTML tags dedicated to represent titles (mainly, `<h1>`) are not always used, or are used several times in the same page. Section 2.1. describes heuristics used to extract the web page title.

We then use conditional random fields (CRFs) in order to extract all dates related to the web page itself (creation date, last update date and current date), as opposed to link-related or content-related dates (Section 2.2.). Finally, we parse the date texts; from this set, we choose a date that we consider as the DCT (Section 2.3.). We analyze dates at the day level, even when more fine-grained time information are provided in the text.

This workflow is illustrated in Figure 2.

2.1. Extracting Article Title

We use structured-based, hand-defined, prioritized rules for extracting article title from the web page. These rules are the following, from highest to lowest priority level:

1. Content of tag `<h1>`, if only one such tag is present in the document.
2. Content of any tag, if it is the longest string in the web page that is included in the HTML header `<title>` tag, and length higher than 15 characters.
3. Content of tag `<h2>`, `<h3>` or `<h4>`, if only one such tag is present in the document.
4. Content of any tag, if the `id` or `class` attributes match language-dependent regular expressions (for example, "`.*title.*`", "`.*headline.*`"), and does not match another set of regular expressions (such as "`.*menu.*`" or "`.*nav.*`", that would probably represent a menu title rather than the main content title).

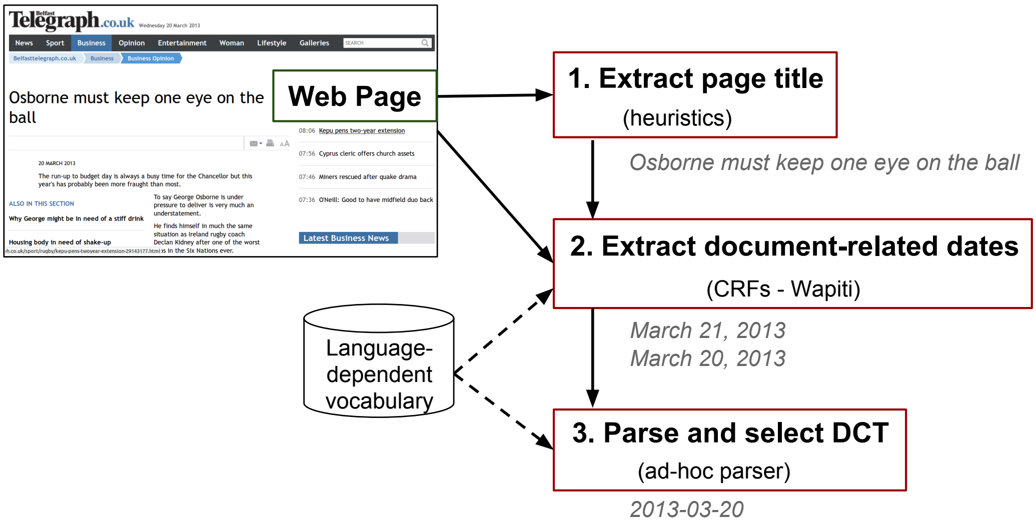

Web Page
Osborne must keep one eye on the ball
1. Extract page title
(heuristics)
Osborne must keep one eye on the ball
2. Extract document-related dates
(CRFs - Wapiti)
March 21, 2013
March 20, 2013
Language-dependent vocabulary
3. Parse and select DCT
(ad-hoc parser)
2013-03-20

Figure 2: DCTFinder workflow.

2.2. Extracting Document-Related Dates with CRFs

As already stated, the difficulty in extracting the page creation date is not to detect dates, but to choose the date that represents the DCT. Context and structure around dates are the main clues in this task. For this reason, conditional random fields (CRFs) are particularly suitable for our problem (Lafferty et al., 2001). This statistical modelling method is often applied in pattern recognition and structured prediction tasks.

After a quick study achieved on a development set, we chose to build a classifier extracting indifferently the date of the day (“now”), the creation date and the last update date. The reason for this choice is that these three dates can appear in very close contexts and structure, and that it seems difficult to differentiate them. On the other hand, it is easy to select the DCT from these three dates after this classification step: it is simply the oldest one.

In CRFs, the training and test sets consist in the textual, tokenized content of the page, where each word and each HTML tag is associated to a number of features. Then, a feature template specifies which features will be used by the learning algorithm, as well as the relative position of the considered token feature. For example, in “*published on Apr. 8, 2013*”, we can learn that “*Apr*” (position n) is probably part of the DCT content because:

1. it is itself a month name;
2. it follows the word “*published*” at position $n-2$;
3. it is followed by a number and a year pattern at positions $n+1$ and $n+2$.

CRFs make this modelisation possible. One can also specify whether this template applies to the current class itself (“unigram”) or to the association between the current class and the previous one (“bigram”). Finally, all numeric values must be discretized.

We used the CRF toolkit Wapiti (Lavergne et al., 2010). Implemented features and templates are described extensively in this section; there are composed of lexical (language-dependent) features, such as date detection patterns or specific vocabulary, as well as structural features, such as word number in a tag, number of dates in the article, distance from title.

2.2.1. Lexical (language-dependent) Features

DETAILED_VOC. Date vocabulary and patterns (*month*, *year*, *day*, *full date* (such as “*04/08/2013*”), *date zones* (GMT, CET, etc.), *time*. Date triggers (“*published*”, “*created*”, “*release*”, “*on*”...) and anti-triggers (“*comments*”...) are also included into DETAILED_VOC.

VOC. Whether the word belongs to a date vocabulary class, such as a month name, a year pattern, a full date or a time zone (boolean value).

FULL_DATE. Whether the word is a full date (boolean value).

2.2.2. Structural Features

WORD_NUMBER. Word number in tag.

POSITION. Relative position in document: first quarter, second quarter, etc.

DATE_ELEM_AROUND. Number of date-related lexical entries around.

DATES_SO_FAR. Number of date patterns met so far in the document.

DATES_IN_ALL. Number of date patterns in the entire document.

DISTANCE_FROM_TITLE. Distance from the title.

DISTANCE_FROM_TRIGGER. Distance from lexical triggers.

DISTANCE_FROM_ANTI_TRIGGER. Distance from lexical anti-triggers.

#	Template		
	Feature	Positions	Type
1	VOC	-2, -1, 0, 1	U
2	DETAILED_VOC	-2, -1, 0, 1	U
3	VOC FULL_DATE	-2, -1 0	B
4	VOC WORD_NUMBER	-1, 0 0	B
5	VOC POSITION	-1, 0 0	B
6	VOC DATE_ELEM_AROUND	-2, -1, 0 0	U
7	VOC DATES_SO_FAR	-1, 0 0	U
8	VOC DATES_IN_ALL	-1, 0 0	U
9	VOC DIST_FROM_TRIGGER	-1, 0 0	U
10	VOC DIST_FROM_ANTI_TRIGGER	-1, 0 0	U
11	VOC DIST_FROM_TITLE	-1, 0 0	U

Table 1: CRF templates used for Wapiti training. The feature names are detailed in Section 2.2.. "Positions" specify the relative positions from the current focusing token, when exploring context. "Type" specifies whether it is a unigram or a bigram template[3].

All numeric values were discretized, using a mapping chosen empirically on a development set. All mappings are described in DCTFinder documentation.

CRF templates used in the evaluation presented in Section 3. are detailed in Table 1.

Headers and scripts are removed from the HTML code before building the CRF tool output, as well as tokens that are not around a numeric value (two tokens before and two tokens after). As dates are rare compared to non-date tokens in a web page, this prefiltering is necessary to prevent the CRF tool from "underfitting" and to predict only non-dates.

2.3. Extracting Document Creation Time

The output of the CRF system is a list of tokens (words and HTML tags), where the tokens belonging to supposed document-related dates are tagged. From this output, parsing the corresponding date is straightforward (with the notable exception of differences between GB and US English date formats, see Section 2.4.). Patterns for years, months, days, etc., described above, are used once again, and strings are replaced by their numeric values (*e.g.* Feb. → 2). When only day and month are provided (for example, *"April 8"*, with no year), the absolute date is deduced from the download date (if specified by the user).

The CRF tool can provide several tagged text spans for the same document. As explained above, these several dates are generally the current date, the DCT or the last update date. Among those dates, we thus select the oldest one as the DCT.

2.4. Language Specificities

For most alphabetic languages, it is not necessary to learn the system again with language-specific annotated pages. Section 3. will show that a simple translation of vocabulary files from English to French leads to good results, even if they may be improved by a language-specific supervision. Different encoding systems are also handled by the system (provided that the encoding is specified in the web page — default is UTF-8).

However, a simple "patch" for English web pages is necessary to improve results. This is because some full date formats are different between pages from United States and pages from other English-speaking countries. For example, "April 8, 2013" is often written *"04/08/2013"* by US writters, and *"08/04/2013"* by others. This can affect the parsing of the dates.

DCTFinder implements two (imperfect) heuristics to handle this issue:

1. If the web page domain name ends by "`.com`", "`.net`", "`.org`", "`.tv`" or "`.us`", then US English is prefered. Otherwise ("`.co.uk`", "`.ca`", "`.nz`", etc.), GB English is prefered.

2. If, given a chosen parser, the parsed date is after the download date (*e.g.*, *"08/04/2013"* parsed as "August 4, 2013" when download date is in April 2013), then we try with the other parser.

However, this difference is still the reason of a significant number of errors.

3. Evaluation

Three different sets of web pages have been annotated:

1. The L3S-GN1 corpus, made of 621 news articles in English, from 408 different web sites[4], published in late 2007 and early 2008. This corpus was initially gathered for training the boilerplate detection tool BoilerPipe (Kohlschtter et al., 2010). Due to several technical issues, we were only able to annotate 567 pages from this set.

2. In order to check that learning from web pages written in 2008 was not a problem (web design knows fashion as every cultural behavior), we also gathered a new dataset of 100 more news articles written in English, published in 2013, coming from 97 different web sites.

3. Finally, we annotated 100 pages in French (from different French-speaking countries), coming from 95 different web sites.

All these datasets and their annotations are made freely available on the tool website.

[3] See `http://wapiti.limsi.fr/` or `http://crfpp.googlecode.com/svn/trunk/doc/index.html` for more details about templates in CRFs.

[4] `http://www.l3s.de/~kohlschuetter/boilerplate/`

Dataset	Title Accuracy	DCT Accuracy
L3S-GN1 (cross-validation)	86.0%	92.4%
English recent dataset	94.0%	90.0%
French recent dataset	88.0%	87.0%

Table 2: Results for title and web page creation time detection.

We used the tool WebAnnotator (Tannier, 2012) for the annotation. This browser extension makes the process easy by keeping the page rendering during annotation.

On the first dataset (L3S-GN1), we performed a 10-fold cross-validation, with 10% of the dataset as development set. Then, the system was learned on the entire L3S-GN1 dataset and applied to the two other datasets.

Table 2 shows the results for title and DCT extraction on these datasets. Results confirm that DCTFinder is accurate enough to be used in NLP applications for web page temporal analysis. We get good results for French dataset, although the model was learned on English documents. However, as multiterm date formats are different in all languages, learning again on language-specific data should improve robustness. The annotation effort is reasonable (less than one hour per 100 web pages).

4. Conclusion

In this paper, we presented DCTFinder, a system extracting the title and the creation date of news web pages. Combined with a web page cleaner such as BTE or BoilerPipe, for extracting the informative content from the page, DCTFinder makes the temporal analysis of web content possible.

With a reasonable error rate, we can bring back the web temporal parsing to the same situation as newswire article parsing, *i.e.* a title, a document creation time and and textual content. This opens the way to large volumes of temporal knowledge that could be helpful in many applications, such as question-answering, information extraction and multidocument text aggregation.

This tool is made freely available for research, as well as all resources (vocabulary and annotated documents) built for training the system, and the trained model itself, at the following URL:

`http://sourceforge.net/projects/dctfinder/`

5. References

Omar Alonso, Ricardo Baeza-Yates, and Michael Gertz. 2007. Exploratory Search Using Timelines. In *SIGCHI 2007 Workshop on Exploratory Search and HCI Workshop*.

Omar Rogelio Alonso. 2008. *Temporal information retrieval*. Ph.D. thesis, University of California at Davis, Davis, CA, USA. Adviser-Gertz, Michael.

Regina Barzilay and Noemie Elhadad. 2002. Inferring Strategies for Sentence Ordering in Multidocument News Summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Angel X. Chang and Christopher Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In LREC2012 (LRE, 2012).

Michele Filannino, Gavin Brown, and Goran Nenadic. 2013. ManTIME: Temporal expression identification and normalization in the TempEval-3 challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Aidan Finn, Nicholas Kushmerick, and Barry Smyth. 2001. Fact or fiction: Content classification for digital libraries. In *Proceedings of the joint DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries*.

Sanda Harabagiu and Cosmin Adrian Bejan. 2005. Question Answering Based on Temporal Inference. In *Proceedings of the Workshop on Inference for Textual Question Answering*, Pittsburg, Pennsylvania, USA, July.

Hyuckchul Jung, James Allen, Nate Blaylock, Will de Beaumont, Lucian Galescu, and Mary Swift. 2011. Building timelines from narrative clinical records: initial results based-on deep natural language understanding. In *Proceedings of BioNLP 2011 Workshop*, BioNLP'11, pages 146–154, Portland, USA. Association for Computational Linguistics.

Nattiya Kanhabua. 2009. Exploiting temporal information in retrieval of archived documents. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, page 848.

Rémy Kessler, Xavier Tannier, Caroline Hagège, Véronique Moriceau, and André Bittar. 2012. Finding Salient Dates for Building Thematic Timelines. In *50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 730–739, Jeju Island, Korea.

Youngho Kim and Jinwook Choi. 2011. Recognizing temporal information in korean clinical narratives through text normalization. *Health Inform Res*, 17(3):150–5.

Christian Kohlschtter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate Detection using Shallow Text Features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM)*, New York, USA.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical Very Large Scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513, Uppsala, Sweden, July. Association for Computational Linguistics.

Hector Llorens, Leon Derczynski, Robert Gaizauskas, and Estela Saquete. 2012. TIMEN: An Open Temporal Expression Normalisation Resource. In LREC2012 (LRE, 2012).

2012. *Proceedings of the Eighth International Language Resources and Evaluation (LREC'2012)*, Istanbul, Turkey, May.

Inderjeet Mani, James Pustejovsky, and Robert Gaizauskas, editors. 2005. *The Language of Time: A Reader*. Oxford University Press.

Jan Pomikálek. 2011. *Removing Boilerplate and Duplicate Content from Web Corpora*. Ph.D. thesis, Masaryk University, Faculty of Informatics, Brno.

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. ISO-TimeML: An International Standard for Semantic Annotation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Estela Saquete, Jose L. Vicedo, Patricio Martínez-Barco, Rafael Muñoz, and Hector Llorens. 2009. Enhancing QA Systems with Complex Temporal Question Processing Capabilities. *Journal of Artificial Intelligence Research*, 35:775–811.

Jannik Strötgen and Michael Gertz. 2012. Temporal Tagging on Different Domains: Challenges, Strategies, and Gold Standards. In LREC2012 (LRE, 2012).

Jannik Strötgen and Michael Gertz. 2013. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298.

Xavier Tannier. 2012. WebAnnotator, an Annotation Tool for Web Pages. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, Turkey, May.

Marc Verhagen, Robert Gaizauskas, Franck Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 - 15: TempEval Temporal Relation Identification. In *Proceedings of SemEval workshop at ACL 2007*, Prague, Czech Republic, June. Association for Computational Linguistics, Morristown, NJ, USA.

M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky. 2010. SemEval-2010 - 13: TempEval-2. In *Proceedings of SemEval workshop at ACL*, Uppsala, Sweden.