

TOB: Timely Ontologies for Business Relations

Qi Zhang¹, Fabian M. Suchanek², Lihua Yue¹, Gerhard Weikum²

¹Computer Science Department, University of Science and Technology of China, Hefei, China

²Max-Planck-Institute for Computer Science, Saarbrücken, Germany

Contact: qzhang@mpi-inf.mpg.de

ABSTRACT

In this paper we present a suite of methods for extracting temporal relations from semi-structured and textual Web sources. We particularly address the needs for building and maintaining business ontologies, where the time aspects of relations between companies, between companies and products, and between companies and customers are important. For example, the date on which a company acquired another company or when a new CEO took over is crucial information for business-intelligence applications. Our methods are geared for extracting business relations and their time information from three kinds of sources: Wikipedia infoboxes, Reuter's news feeds, and news pages provided by Google. All techniques are integrated into the TOB framework for timely business ontologies. Our experiments show that we can achieve fairly high precision for the extracted information.

1. INTRODUCTION

The Web has become a valuable source of facts that can be obtained by information-extraction (IE) methods. There is rich literature on methods based on pattern matching, linguistic analysis, and statistical learning, for extracting entities and relations among entities from Web resources. The value-added information IE can be used to automatically build and maintain ontologies and knowledge bases, as a substrate for entity-search on the Web, and as a general asset for future text-mining tasks. Unfortunately, state-of-the-art IE systems handle time aspects of real world relations very insufficiently. Time properties are important in understanding relations, especially in the business world. For example, when we say that Jack Welch is the CEO of General Electric, this fact was true between 1981 and 2001 but no longer holds today.

In this paper, we introduce the TOB toolkit for automatically building time-annotated business ontologies. We present a suite of methods for extracting temporal relations from semi-structured and textual Web sources. As a result, we can map real-world facts (i.e., instances of binary relations) onto a timeline. In this work, we mainly focus on business relations. But our algorithms can work in other domains as well, such as scientific communities (e.g., the PC chair relation of a conference for a certain year) or sports (e.g., championship winners). Figure 1 shows an example of time-annotated facts.

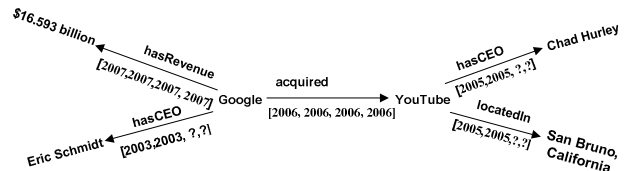


Figure 1. Sample business relation in TOB

It shows the acquisition relation between two companies: Google and YouTube. As shown in the figure, every relation in our system is labeled with time intervals. Based on the extracted and time-annotated facts, we can answer questions like: Who is the CEO of General Electric now? What was Google's revenue in the year when it acquired YouTube?

There are many good IE systems available for extracting binary relations from text documents, such as Gate/Annie [28], Snowball [1], Text2Onto [3], Leila [6], TextRunner [16]. But unfortunately, none of these systems works for temporal relations with satisfactory precision. We propose a suite of methods for extracting and inferring temporal relations from semi-structured and natural-language text. The contributions are:

- We extend an existing IE tool based on dependency-grammar parsing to *extract temporal relations*.
- We introduce a model to represent *underspecified time intervals*, which is particularly designed for Information Extraction.
- We develop new ways of *temporal relation inferencing* to deal with situations where a sentence does not yield the temporal context:

Ontology-level inference allows us to deduce time spans for facts from time spans of entities. For example, if we cannot extract the date when "Google acquired YouTube" but know from the ontology that Google was founded in 1982 and YouTube was founded on February 15, 2005, we can infer that the acquisition must have happened after February 15, 2005.

Page-level inference allows us to instantiate relative time expressions from the date of the page. For example, if a news page says "Google acquired YouTube last Monday" and if we can extract "Wednesday, April 2" as the publication date of the article, we can infer the date of the acquisition.

The paper is organized as follows. Section 2 discusses related work. Section 3 presents the timely ontology model. Section 4 explains our extraction methods. Section 5 explains our inferencing methods. Section 6 presents the results of a small-scale experiment with real-life data from Wikipedia, Reuters, and Google. Our methods achieve remarkably good precision at reasonable recall level.

2. RELATED WORK

Numerous approaches have been proposed to create general-purpose ontologies by automatically extracting facts from text corpora. These approaches use information extraction technologies that combine pattern matching, natural-language parsing, and statistical learning; see [20, 23, 14] for overviews and, for example, [1, 17, 23, 6, 16] for specific techniques and software tools. Important projects on large-scale fact extraction include KnowItAll [4], TextRunner [16], Libra [22], Avatar [24], and Cimple [13, 18]. KnowItAll aimed at extracting and compiling instances of a given set of unary and binary predicate instances on a very large scale – for example, as many soccer players as possible or almost all company/CEO pairs from the business world. TextRunner has the even more ambitious goal of extracting all instances of all meaningful relations from Web pages, a paradigm referred to as “machine reading” [9]. Libra and Avatar are comprehensive tool suites for extracting records from Web pages, as an asset towards entity search over the Web (see also [19] for a similar approach to the search task). Cimple is a framework for information extraction and integration that has been successfully utilized to build and maintain the DBLife portal.

A recent line of research is to automatically derive facts from the automatically understandable parts of Wikipedia. This direction includes DBpedia [15], Kylin [12], YAGO [7, 29], and the work by [5, 30]. The DBpedia project was started by extracting facts from the infoboxes of particular types of Wikipedia articles (e.g., on people, cities, companies, music bands, etc.). YAGO exploits both infoboxes and the Wikipedia category system and integrates the extracted facts with the WordNet thesaurus in a consistent manner. YAGO uses various forms of disambiguation and consistency checking for high accuracy. Kylin has used extraction techniques on infoboxes, similar to those of DBpedia, but additionally applies powerful learning techniques to automatically fill in missing values in incomplete infoboxes.

There is also related research on time aspects in document retrieval and summarization [8, 10, 11, 29]. [10] introduced ways to normalize relative temporal phrases. Our approach differs from these works in two aspects. First, we do not consider time annotations as isolated entities but as an integral enhancement of a primary fact. Second, we combine techniques for publication-date detection with relative time normalization, thus strengthening the extraction accuracy. An overview of NLP (Natural Language Processing) techniques for mining temporal information from texts is given in [21]. Most NLP-centric work addresses ordering of events, which is not related to our problem.

3. REAL TIME BUSINESS ONTOLOGY

3.1 Basic Ontology Model

The ontology model of TOB is based on the YAGO model [7], which in turn is a variant of RDF. In YAGO, all objects are represented as entities, such as companies, people, and products. *Facts* are binary relations between two entities. For example, to state that Elvis won a Grammy Award, we connect the two entities `Elvis` and `Grammy_Award` with the `hasWonPrize` relation:

```
Elvis Presley hasWonPrize Grammy_Award
Classes (i.e., entity types) and relations are also entities. For example, the subclass-relation between the class singer and class person can be represented as:
```

```
singer subclassOf person.
```

Every fact is also an entity and gets assigned a fact identifier. Some facts may involve more than two entities, for example, the fact that Google acquired YouTube on October 9, 2006. In order to support such n-ary relations in a binary-relation (or RDF subject-property-object triple) model, YAGO splits the n-ary fact into a primary fact and associated facts. The associated facts are represented by relations that hold between the identifier of the primary fact and the remaining arguments. For the example, if the fact `Google acquired YouTube` has the identifier #1, we can represent the remaining part of the ternary relation as:

```
#1 happenedOn 2006-10-09
```

3.2 Time-Enhancement for the Ontology

The key characteristic of TOB in going beyond YAGO is that every fact should have a time interval indicating when the fact happens. Our model uses *basic time intervals* as elementary building blocks. A basic time interval is a calendar date (e.g. 2008-06-03), a year (2008), a century (15th century BC) or any other time literal. We propose to model the time range of a fact f by 4 relations, which each hold between the identifier of f , id_f , and a basic time interval:

```
id_f startsAfter t1
id_f startsBefore t2
id_f endsAfter t3
id_f endsBefore t4
```

This representation means that f starts after the start of t_1 , it starts before the end of t_2 , it ends after the beginning of t_3 and it ends before the end of t_4 . We use basic time intervals instead of time points because in many cases, we do not have the exact information on when the fact starts and ends. The basic time intervals are constrained as follows, where *start* and *end* denote the starting and ending times of intervals, respectively:

$$start(t_1) < end(t_2); start(t_2) < end(t_3); start(t_3) < end(t_4);$$

We call the quadruple of t_1, t_2, t_3, t_4 a *fuzzy time range* and abbreviate it as follows in this paper:

$$f: [t_1, t_2, t_3, t_4]$$

If we know the start time of f and the end time of f , we set $t_1=t_2$ and $t_3=t_4$. If one of the basic time intervals is unknown (and thus one of the time facts is missing), we write a question mark “?” to substitute the unknown bound. The question mark can be seen as the basic time interval that represents the infinite time interval from the far past to the far future.

For example, assume we know that Jack Welch became CEO of General Electric at some time between 1980 and 1982

(inclusively). He retired between 2000 and 2001. Then the fact “Jack Welch was CEO of GE” can be expressed as follows:

```
id1 Jack Welch isCEO General Electric
id2 id1 startsAfter 1980
id3 id1 startsBefore 1982
id4 id1 endsAfter 2000
id5 id1 endsBefore 2001
```

An abbreviation of the fuzzy time range is: [1980, 1982, 2000, 2001]

3.3 Work Flow of TOB

Figure 2 shows the system architecture and workflow of the TOB toolkit. The main building blocks are explained in the subsequent sections.

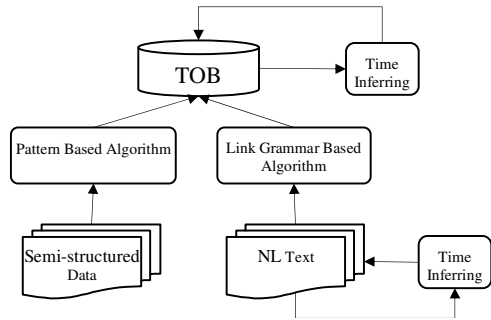


Figure 2. Work flow of TOB.

4. RELATION EXTRACTION

4.1 Pattern Based Relation Extraction

In contrast to the text-oriented pattern matching by prevalent tools like Gate/Annie [28], Snowball [1], or TextRunner [16], our pattern-based extraction focuses on structured and semi-structured data, a typical and most prominent example being infoboxes in Wikipedia. This approach has been used in prior work (by YAGO [7] and DBpedia[15]), but in TOB we not only deal with basic relations (e.g., CEOs of companies), but also aim to associate the extracted facts with time annotations. Our general rationale for addressing these kinds of semi-structured sources is to bootstrap the ontology with high-quality facts that are valid with very high confidence.

For example, here is a piece of code from the infobox in the Wikipedia article about Google in the Wikipedia Markup Language:

```
num_employees = 16,805 ([[Dec31]] [[2007]])
```

Our TOB tool uses pattern matching to extract the following fact:

```
Google hasEmp 16,805 [2007-12-31, 2007-12-31,
2007-12-31, 2007-12-31]
```

4.2 Link Grammar Based Relation Extraction

For high coverage (recall) of business facts we need to consider textual Web pages, most notably news pages. We use E-Leila, a novel extension of Leila [6], to extract facts from texts.

4.2.1 Leila

Leila builds on a deep parser for natural-language sentences based on a link grammar [25] (aka. dependency grammar). This yields a rich, graph-based feature representation of a candidate sentence, and the actual fact extraction uses statistical learning (e.g., SVMs) on this representation. In the graph representation of a sentence, a relation r is reflected by the linkage (path) between two named entities (proper nouns). A *pattern* is the linkage in which two entities (source and destination) have been replaced by placeholders, as shown in Figure 3. A *bridge* is the shortest path from one placeholder to the other. A pattern *matches* a linkage (of a newly seen test sentence) if the bridge of the pattern appears in the linkage.

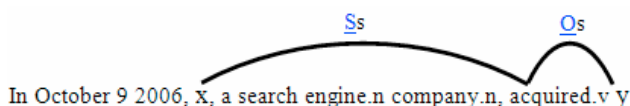


Figure 3. A sample pattern.

The example in Figure 4 is matched by the pattern of Figure 3.

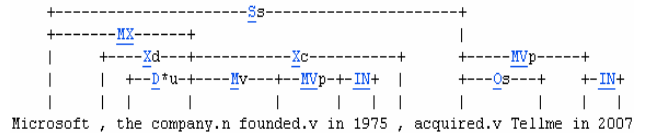


Figure 4. Matched example

4.2.2 E-Leila

In [6], Leila outperformed other pattern-based methods in terms of extraction accuracy. However, Leila supports only binary-relation extraction; thus, it is of limited use for TOB.

We have developed an extension of Leila, coined *E-Leila*, which can deal with ternary and quaternary relations to capture facts together with their time intervals. The algorithm is outlined in Figure 5.

Algorithm ExtendedLeila

Input: A sentence *Sent*

Output: A list of facts.

```
parsedGram ← LinkGrammarParser(Sent)
factList ← Leila(parsedGram)
dateList ← DateRecognition(Sent)
for each fact in factList, do
    verb ← GetVerb(parsedGram, fact)

    for each date in dateList, do
        prep ← GetPreposition(date, Sent)
        if HasLinkage(pre, verb) is true, then
            newFactList ← GetTimeRelation(pre, Sent)
            factList ← factList + newFactList

return factList
```

Figure 5. E-Leila pseudocode

The *DateRecognition* function takes a sentence as input, and outputs a date list which includes all the date instances in the sentences. Here, dates include years and months. We recognize dates mainly by patterns.

The *GetVerb* function detects the main verb in the relation fact, such as “acquired” for the “Google acquired YouTube” example.

The *GetPreposition* function identifies the preposition that precedes the date entity. In our algorithm, we only deal with date entities that indeed have a preposition.

The *HasLinkage* function checks if there is a direct linkage between the preposition and the primary verb. If there is such a linkage, we have found a temporal property of the primary fact.

The *GetTimeRelation* composes new temporal facts from the primary fact id and the dates that have a linkage to the primary verb. It constructs different time facts for different prepositions (see Table 2).

Table 1. Prepositions and the time intervals for time t

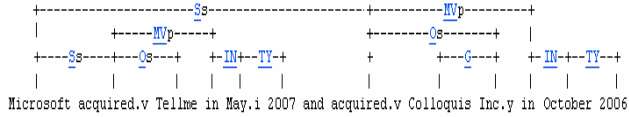
Preposition	Time Range
in, on, at, during	[t, t, t]
since	[t, t, ?, ?]
after	[t, ?, ?, ?]
before	[?, ?, ?, t]
till, until, by	[?, ?, t, t]

In the following we show an example of the algorithm:

Input:

Microsoft acquired Tellme in May 2007 and acquired Colloquis Inc in October 2006.

Result of Link Parser:



Result of Leila:

Microsoft acquired TellMe
 Microsoft acquired Colloquis_Inc

Result of Extended Leila:

Microsoft acquired TellMe [2007-05,2007-05,2007-05,2007-05]
 Microsoft acquired Colloquis_Inc [2006-10, 2006-10,
 2006-10, 2006-10]

5. TEMPORAL RELATION INFERENCE

Often, there is no temporal information in the same record or sentence that contains the primary fact. Our goal in TOB is to have every fact associated with a time interval. To this end, we have developed new ways of inferring temporal annotations from context information: the existing ontology itself and the Web page where we found the primary fact.

5.1 Ontology Level Inferencing

Most entities have a life span in which they participate in events. Companies, for example, do not exist prior to their foundation or after their dissolution. Formally, the *life span of an entity e* is a fuzzy time range that indicates the time in which *e* can participate in facts. The life span can be computed as follows, depending on the type of *e*:

- if *e* is a fact, its life span is as defined in Section 3.
- if *e* is an individual entity such as company, person, etc., the start date is the foundation date of the company or the birth date of the person. The end date is the dissolution date of the company (e.g., bankruptcy or acquisition by another company) or the death date of the person.
- in all other cases (e.g. if the entity is a Website address), the life span is [?, ?, ?, ?].

A fact can only hold during the life spans of its arguments. More formally, if the fact is about the two entities *e*₁ and *e*₂, and *e*₁ has life span *T*₁ and *e*₂ has life span *T*₂, we may narrow down the fuzzy time range *T* of the fact as follows:

$$T := T \nabla T_1 \nabla T_2$$

Here, ∇ is the *time inferencing operator*, defined as follows for two fuzzy time ranges $T_1=[t_1, t_2, t_3, t_4]$, $T_2=[t_1', t_2', t_3', t_4']$:

$$T_1 \nabla T_2 := [laterStart(t_1, t_1'), ?, ?, earlierEnd(t_4, t_4')]$$

The functions *laterStart* and *earlierEnd* return the basic time interval that has a later start and an earlier end, respectively. We use the infinite time interval “?” in the *startsAfter* and *endsBefore* component, because without other information, the end bound of the start time and the start bound of the end time of the fact are unknown.

Take as an example the fact $f = \text{Google hasCEO Eric Schmidt}$ with an unknown time interval. We know that the life span of Google is [1998-09-07, 1998-09-07, ?, ?] and the life span of Eric Schmidt is [1955-04-27, 1955-04-27, ?, ?]. So the time interval of *f* can be narrowed down as:

$$[?, ?, ?, ?] \nabla [1998-09-07, 1998-09-07, ?, ?] \nabla [1955-04-27, 1955-04-27, ?, ?] = [1998-09-07, ?, ?, ?]$$

This may leave the time annotation underspecified, but it constrains the possible dates and is thus very valuable for further inferencing and for querying the ontology at a later stage. Note that the ∇ operator is different from the fuzzy time range intersection \cap , which could be defined in a similar way. \cap could be used, e.g., to compute a more precise fuzzy time range from two fuzzy time ranges for the same fact.

5.2 Page Level Inferencing

News pages contain many relative temporal phrases, such as “today”, “last Monday”, “this year”, etc. This makes the extraction of proper time points or time intervals much harder.

In TOB, we aim to normalize these relative temporal phrases by inferencing from the page context. We do this in two steps:

1. **Publication date identification:** This is the date when the page was published. It is key feature for news articles, and indeed often detectable for news and Web pages of well-run organizations (as opposed to arbitrary Web pages).
2. **Relative temporal phrases normalization:** The relative temporal phrases are translated into absolute time intervals by inference from the publication date.

5.2.1 Publication Date Identification

There are three sources where the publication date may appear:

- **URL:** The publication date may appear in the URL, such as for example in <http://politicalticker.blogs.cnn.com/2008/04/02/obama-wants-gore-on-his-team/>
- **Metadata:** The date also may appear in the metadata, for example: `<meta name="pub_date" content="20070315"/>`
- **Main Text:** When neither URL nor metadata provides a date, we analyze the text of the page. There are often good cues for the position of a publication date:
 - It is between the title and the main text.
 - It follows keywords like: “Published”, “Posted”, or “Publication Date”.
 - It follows the author name.

We use a rule-based algorithm based on these cues to identify the publication date.

5.2.2 Relative Temporal Phrase Normalization

We normalize the relative temporal phrases to absolute dates by adding or subtracting time spans from the current date. Currently, our system supports the phrases shown in Table 2. This part of our work was inspired by [10].

Table 2. Example for Relative Temporal Normalization (assuming the current date is 2008-03-03)

	Example	
Temporal Type	Before Normalization	After Normalization
Day	Tomorrow	2008-03-04

Week	Last week	2008-w09
Month	April	2008-04
Year	Three years ago	2005

6. EVALUATION

6.1 Setup

We prepared different corpora for experimental evaluation:

- **Wikipedia companies:**
We compiled 350 Wikipedia articles about US companies that are listed in http://en.wikipedia.org/wiki/List_of_United_States_companies. These pages are used to test our pattern-based extraction algorithm.
- **Reuters company descriptions:**
We downloaded 276 Reuters company description pages, listed at <http://stocks.us.reuters.com/stocks/lookup.asp>. We use this dataset to test the link-grammar-based relation extraction algorithm.
- **Google News pages:**
To test our page-level temporal inferencing methods, we collected 438 news pages from the Google News Archive with publication dates different from the current date.

6.2 Results

6.2.1 Results on pattern-based extraction

The 350 Wikipedia company articles included 274 infoboxes. We manually chose 29 attributes which are common in company infoboxes, like “*key people*”, “*company type*”, “*foundation*”, “*headquarters*”, “*net income*” etc.

From these 274 infoboxes, we extracted 4520 facts. There were 439 facts with a time property. There were also 261 infoboxes that contained the foundation date of the company.

We manually assessed the precision of the results. The manual evaluation is unobjectionable, because there is hardly any ambiguity in the business facts (unlike in relevance assessments for query results, e.g., where subjectivity plays a role). Since we could not evaluate all extracted facts, we evaluated a random sample for each relation type. The results are shown in Table 3. We computed the Wilson confidence interval for $\alpha=0.05$ and increased the sample size until the interval shrank below $\pm 10\%$ (unless there were not enough facts for a specific relation type). The last row of the table shows the average precision over all relation types. These results prove that our method achieves a very high precision.

Table 3. Precision for extraction from infoboxes

	Relation Type	#Eval	#Correct	Precision
1	hasStockSYMBOL	100	100	98 \pm 2%
2	hasSubsidy	56	50	87 \pm 8%
3	hasDivision	46	43	90% \pm 8%
4	hasEssets	5	5	78% \pm 22%
5	hasProduct	104	77	73% \pm 8%
	...			
10	hasFullName	120	119	98 \pm 2%
11	hasNumOfEmployee	114	110	95 \pm 4%

	...			
20	hasStockMarket	126	126	99% \pm 1%
21	hasIndustry	103	96	92% \pm 5%
22	bornIn	101	94	91% \pm 5%
	...			
27	hasWebsite	94	93	97% \pm 3%
28	hasEquity	5	5	78% \pm 22%
29	hasSlogan	96	84	86% \pm 7%
Total		1608	1540	96%

6.2.2 Results on link-grammar-based extraction

We applied our E-Leila tool to the 276 Reuter company description pages. We focused on the relation of “acquisition” (which are ample in the Reuters data but would be very sparse in Google News). We extracted the facts about business acquisitions manually from the corpus. There are 304 acquisition instances in the corpus; 300 instances have time annotations. The results of this experiment are shown in Table 4. Our methods performed very well in terms of precision, on both extracting primary facts and extracting the time intervals using E-Leila. Recall was about 30%; this may not appear impressive, but high-precision information extraction usually and often inevitably has relatively low recall as its input may include arbitrarily difficult natural-language sentences. Our recall figure is in line with results on general-purpose IE tools on simple relations.

Table 4. Result of relation extraction on Reuters pages

System	Total	Found	Correct	Error	Precision	Recall
Leila	304	110	102	8	91% \pm 5%	34% \pm 5%
E-Leila	300	95	89	6	92% \pm 5%	30% \pm 5%

For a few input sentences, we could extract the primary fact, but not the time intervals. These (relatively few) cases were caused by somewhat informal, colloquial language use, such as the following:

1. There is no preposition preceding the date phrase. For example: “November 16, 2007, Ultralife Batteries Inc. completed the acquisitions of Stationary Power Services, Inc.”
2. The preposition is not directly connected with the date. For example: “During the year ended December 31, 2007, Thomson Corp completed the acquisition of Deloitte Tax LLP Property Tax Services.”

6.2.3 Results on relative temporal normalization

We annotated the 438 Google News Archive manually with their true publication date. Then we ran our publication-date recognition algorithm on them. The result is shown in Table 5:

Table 5. Publication-date recognition result

#Eva	Correct Number	Incorrect Number	Precision
438	335	103	76% \pm 4%

Due to the various page styles in news pages, our precision is not very high but still fairly good. False positives result from some

pages without proper publication date in the considered context and from pages that have posting dates for comments.

After the publication-date recognition, we ran our relative temporal phrase normalization algorithm. We manually checked the results: there were 1407 instances correctly normalized, and there were 171 instances which our algorithm handled incorrectly. Thus, the precision is about 89%.

7. CONCLUSION

In this paper, we presented the TOB toolkit for automatically building time-annotated business ontologies. In contrast to the ample prior work on general-purpose information extraction, our methods are particularly geared for extracting temporal relations from semi-structured and textual Web sources and perform much better than standard methods. Furthermore, we have developed new ways of temporal relation inferencing for facts without time intervals. Our experiments have shown that we can achieve fairly high precision for the extracted information.

For future work, we aim to integrate our system with YAGO to enlarge the YAGO ontology. Meanwhile, we will study the problem of informal, colloquial language input which E-Leila currently couldn't process.

8. REFERENCES

- [1] E. Agichtein and L. Gravano. *Snowball*: extracting relations from large plain-text collections. In *ACM 2000*, pages 85–94.
- [2] Sergey Brin. Extracting patterns and relations from the World Wide Web. In *Selected papers from the Int. Workshop on the WWW and Databases*, pages 172–183, London, UK. Springer-Verlag.
- [3] P. Cimiano and J. Völker. Text2onto - a framework for ontology learning and data-driven change discovery. In A. Montoyo, R. Munozand, and E. Metais, editors, *Proc. of the 10th Int. Conf. on Applications of Natural Language to Information Systems*, pages 227–238, Alicante, Spain.
- [4] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall (preliminary results). In *WWW 2004*, pages 100–110.
- [5] Simone Paolo Ponzetto, Michael Strube: Deriving a Large-Scale Taxonomy from Wikipedia. *AAAI 2007*: 1440-1445.
- [6] Fabian M. Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents. In *SIGKDD 2006*.
- [7] Fabian M. Suchanek, Gjergji Kasneci and Gerhard Weikum " YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia" *WWW 2007*, ACM Press, pp. 697–706.
- [8] Allan, J., Papka, A., & Lavrenko, V. . On-line new event tracking. In J. Zobel (Ed.), *ACM SIGIR 1998*.
- [9] Oren Etzioni, Michele Banko, Michael J. Cafarella: Machine Reading. *AAAI 2006*
- [10] Koen, D. B., & Bender, W. Time frames: temporal augmentation of the news. *IBM Systems*, 39(3&4), 597–616.
- [11] Swan, R., & Allan, J. Automatic generation of overview timelines. In P. Ingwersen (Ed.), *SIGIR 2000*.
- [12] Fei Wu, Daniel S. Weld: Autonomously semantifying wikipedia. *CIKM 2007*: 41-50
- [13] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, and W. Shen. Community information management. In *IEEE Data Engineering Bulletin*, Special Issue on Probabilistic Databases, volume 29, 2006.
- [14] Eugene Agichtein, Sunita Sarawagi, Scalable Information Extraction and Integration, Tutorial, *KDD 2006*,
- [15] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, Zachary G. Ives: DBpedia: A Nucleus for a Web of Open Data. *ISWC/ASWC 2007*.
- [16] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, Oren Etzioni: Open Information Extraction from the Web. *IJCAI 2007*: 2670-2676
- [17] Razvan C. Bunescu, Raymond J. Mooney: Learning to Extract Relations from the Web using Minimal Supervision. *ACL 2007*
- [18] Fei Chen, AnHai Doan, Jun Yang, Raghu Ramakrishnan: Efficient Information Extraction over Evolving Text Data. *ICDE 2008*.
- [19] Tao Cheng, Xifeng Yan, Kevin Chen-Chuan Chang: EntityRank: Searching Entities Directly and Holistically. *VLDB 2007*: 387-398
- [20] William Cohen, Andrew McCallum, Information Extraction and Integration: an Overview, Tutorial, *KDD 2004*
- [21] P. Cimiano and J. Vlker. Text2onto - a framework for ontology learning and data-driven change discovery, 2005.
- [22] Zaiqing Nie, Ji-Rong Wen, Wei-Ying Ma: Object-level Vertical Search. *CIDR 2007*: 235-246
- [23] Hamish Cunningham: Information Extraction, Automatic. in: *Encyclopedia of Language and Linguistics*, 2005.
- [24] T. S. Jayram, Rajasekar Krishnamurthy, Sriram Raghavan, Shivakumar Vaithyanathan, Huaiyu Zhu: Avatar Information Extraction System. *IEEE Data Eng. Bull.* 29(1): 40-48 (2006)
- [25] Daniel Sleator, Davy Temperley: Parsing English with a Link Grammar. Technical Report CMU-CS-91-196, October 1991, <http://www.link.cs.cmu.edu/link/>.
- [26] Steffen Staab, Rudi Studer: *Handbook on Ontologies* Springer 2004.
- [27] Inderjeet Mani: Recent developments in temporal information extraction. *RANLP 2003*: 45-60
- [28] <http://gate.ac.uk/ie/annie.html>.
- [29] PJ Kalcynski & A Chou, Temporal document retrieval model for business news archives *Information Processing and Management: an International Journal archive*. Volume 41, Issue 3 (May 2005)
- [30] M Hepp, K Siorpaes & D Bachlechner, Harvesting Wiki Consensus: Using Wikipedia Entries as Vocabulary for Knowledge Management, *IEEE INTERNET COMPUTING*, Vol. 11, No. 5, pp. 54-65 , Sept-Oct 2007