Terminology Evolution in Web Archiving: Open Issues

Nina Tahmasebi
L3S Research Center
Appelstr. 9a
Hannover, Germany
tahmasebi@L3S.de

Tereza Iofciu
L3S Research Center
Appelstr. 9a
Hannover, Germany
iofciu@L3S.de

Thomas Risse
L3S Research Center
Appelstr. 9a
Hannover, Germany
risse@L3S.de

Claudia Niederée
L3S Research Center
Appelstr. 9a
Hannover, Germany
niederee@L3S.de

Wolf Siberski
L3S Research Center
Appelstr. 9a
Hannover, Germany
siberski@L3S.de

ABSTRACT

The correspondence between the terminology used for querying and the one used in content objects to be retrieved, is a crucial prerequisite for effective retrieval technology. However, as terminology is evolving over time, a growing gap opens up between older documents in (long-term) archives and the active language used for querying such archives. Thus, technologies for detecting and systematically handling terminology evolution are required to ensure "semantic" accessibility of (Web) archive content on the long run. As a starting point for dealing with terminology evolution this paper formalizes the problem and discusses issues, first ideas and relevant technologies.

Categories and Subject Descriptors

H.3.6 [**Library Automation**]: Large text archives; H.3.1 [**Content Analysis and Indexing**]: Linguistic processing

General Terms

Web Archives, Terminology Evolution, Semantics, Information Extraction

1. INTRODUCTION

Due to the central role that the World Wide Web plays in nearly all areas of today's life, its continuous growth, and its change rate, adequate Web archiving has become a cultural necessity [14] in preserving knowledge. Ensuring archival of its content - which is a complex task by itself - is just the first step toward "full" content preservation. It also has to be ensured that content can be found and interpreted on the long run.

This type of *semantic* accessibility of content suffers due to changes in language over time, especially if we consider time frames beyond ten years. Language changes are triggered by various factors including new insights, political and cultural trends, new legal requirements, high-impact events, etc. As an example consider the name of the city Saint Petersburg: This Russian city was founded in 1703 as "Sankt-Piter-Burh" and soon after renamed to "Saint Petersburg". From 1914-1924 it was named "Petrograd" and afterwards "Leningrad". Since 1991 the name changed back to "Saint Petersburg". Evolution of terms is of course not restricted to location names and the terminology change rate clearly depends on the domain of discourse.

Due to this terminology development over time, search with standard information retrieval techniques, using current language or terminology will not be able to find all relevant content created in the past, when other terms were used to express the sought content.

Interfaces for accessing Web archives such as WERA[1], which are under development, are based on traditional information retrieval methods. These approaches work quite well with current web archives as they date back only a couple of years. The problem of *terminological invisibility of content* will only arise for later generations of users.

For keeping Web archives semantically accessible it is necessary to develop methods for automatically dealing with terminology evolution. This includes the detection of terminology evolution as well as ways to integrate the knowledge about terminology evolution into time-aware retrieval approaches, such as the one presented in [4]. The query "Saint Petersburg" could, for example, be expanded with the right terms for the different periods when querying an archive ("Saint Piter Burh" $\xrightarrow[1703]{}$ "Saint Petersburg" $\xrightarrow[1703-1914]{}$ "Petrograd" $\xrightarrow[1914-1924]{}$ "Leningrad" $\xrightarrow[1924-1991]{}$ "Saint Petersburg").

Adequately dealing with terminology evolution requires the consideration of the linguistic and of the semantic layer: in addition to emerging and vanishing terms, it is exactly the

[1]Web ARchive Access: `http://archive-access.sourceforge.net/projects/wera/`

change in the mapping between language (terms used) and concepts (intended meaning) that constitutes terminology evolution. In this paper we present first ideas on how to deal with this phenomenon, including a model for terminology evolution and a review of technologies that can be exploited and combined to detect these changes.

The rest of the paper is structured as follows. As a foundation, Section 2 introduces the different types of terminology evolution that can be identified. Section 3 is dedicated to framing the problem of terminology evolution in a more formal way. First ideas toward a method for automatically detecting terminology evolution together with related work and technologies that could be adopted for this purpose are discussed in Section 4. The paper concludes with plans for future work in realizing a solution for detecting terminology evolution in Web archives.

2. CAUSES OF TERMINOLOGY EVOLUTION

In the above terminology evolution example on "Saint Petersburg" the renaming of the city was caused by political and historical changes. However, there are various other causes for terminology evolution to occur such as cultural and scientific evolution, cultural inter-exchange, political and economical events. The following list of examples illustrates the variety of causes for terminology evolution.

- Neologisms - words borrowed from other languges, e.g. German words in English: *hamburger, fest, muesli*
- Politicall correct terminology
  - *Chairman* is now *Chair* or *Chairperson*
  - *Fireman* is now *Fire fighter*
- Branding, from company name to concept
  - *Google* - "to *google*" is now used for searching on the Web
  - *Jacuzzi* used for whirlpool baths
- Economical changes, for example in company names, e.g. *Spin-Offs:* O2 (British Telecom), Infineon (Siemens)
- Political renames, e.g. *Eastern German States:* from *GDR* and then *New States*

In addition there are changes in the word etymology where the cause of the change is not easily identified:

- *Awesome* previously only meant *terrifying*, and now its meaning also includes *amazing*
- *Bastard* previously only meant *illegitimate child* and now also means *a disagreeable person*

In spite of the different causes for terminology evolution, the essence of terminology evolution can be captured in changes in the relationship between the language layer (terms used) and the concept layer (intended meaning). At one point of time (or more precisely for a period of time) a term is associated with one or more concepts, which defines its meaning(s). Over time, use and meaning of terms may change. This means that the relationships between terms and concepts may change as well as the concepts themselves. In addition new terms may come up and terms may become more or less popular or may vanish completely from daily use. This relationships are considered in a more formal way in the next section.

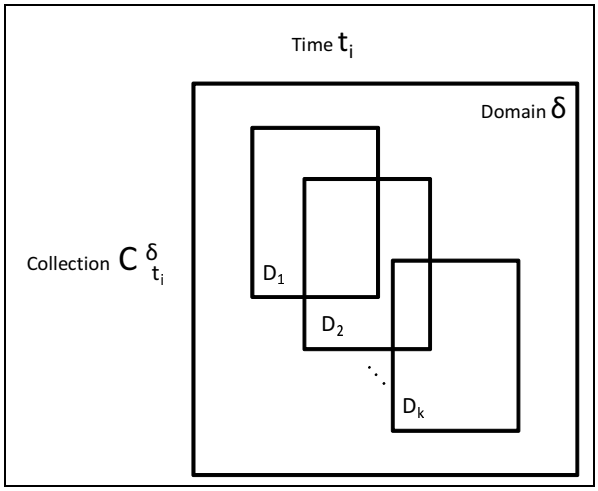

Figure 1: A collection taken in domain δ captured in time interval $[t_{i-1}, t_i]$.

3. A MODEL FOR TERMINOLOGY EVOLUTION

The problem of automatically detecting the terminological evolution of a term can be split into two different sub problems. First we need to identify and represent the relation between terms and their intended meanings (concepts) at a given time. We call such a representation *terminology snapshot*. Such a snapshot is always based on a given document collection. Second, we need to perform a fusion of different terminology snapshots, by identifying the relations between their respective concepts.

3.1 Terminology Snapshots

We start with with some basic definitions that we use in the sub-sequent problem abstractation.

Collections - A collection $C^{\delta}_{t_i}$ as seen in Figure 1 is a set of documents D taken from a domain δ in the time interval $[t_{i-1}, t_i]$, where $i = 1, \ldots, N$.

Terms - Let W be the complete universe of terms in the given language. Each document $D_j \in C^{\delta}_{t_i}$ contains a set of terms $w \in W^{\delta}_{t_i}$. The set $W^{\delta}_{t_i} \subseteq W$ is domain specific and contains all terms ever used in domain δ until time t_i. W^{δ} is unknown for a domain δ, but can be approximated using the given collections. We define W'^{δ} at time t_0 to be the empty set, $W'^{\delta}_{t_0} = \emptyset$ and $W'^{\delta}_{t_i} = W'^{\delta}_{t_{i-1}} \bigcup terms(C^{\delta}_{t_i})$ for $i = 1, \ldots, N$, where $terms(C^{\delta}_{t_i}) = \{w : \exists D\ w \in D \wedge D \in C^{\delta}_{t_i}\}$.

Concepts - To represent the relation between terms and

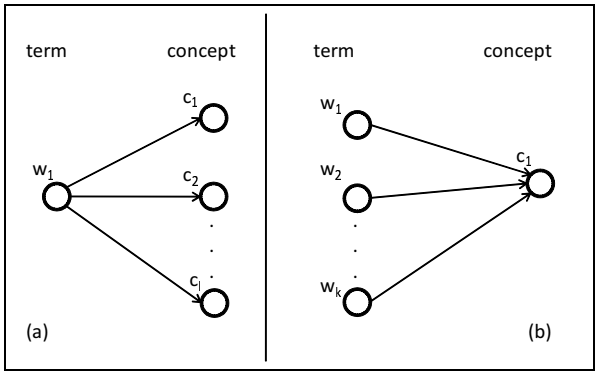

Figure 2: Term relationships without time dimension. (a) One term pointing to several concepts. (b) Several terms pointing to one concept.

their meanings we introduce the notion of *concept* and represent meanings as connections between term and concept nodes in a graph. Let $\mathcal{C}$ be the universe of all concepts. The semantics of a term $w \in W_{t_i}^{\delta}$ is represented by connecting it to its concepts, as shown in Figure 2a. The edges between terms and concepts inherit the time annotation from the collection on which the terminology snapshot is based. For every term $w \in W_{t_i}^{\delta}$, at least one term-concept edge has to exist.

When a word points to several concepts, as can be seen in Figure 2a, this can be candidates for lexical ambiguity [19]. In Figure 2b we can see how synonyms would be represented in a graph. Here we would consider $w_1, \ldots, w_k$ to be synonyms.

Note that we use the linguistic terminology in a rather broad sense, e.g., we also allow to represent individual instances such as a city as concept nodes and view different names for such an instance as synonyms, as in the *Saint Petersburg* example.

We introduce the function ϕ as representation of these term-concept relations as

$$\phi : W \times T \to (W \times \mathcal{P}(\mathcal{C} \times \mathcal{P}(T))) \tag{1}$$
$$(w, t) \mapsto (w, \{(c_1, \{t\}), \ldots, (c_n, \{t\})\})$$

where $w \in W$, $t \in T$ and for all $i = 1 \ldots n$: $c_i \in C$. $\mathcal{P}$ denotes a power set, i.e. the set of all subsets. Although ϕ generates only one timestamp for each term-concept relation, we introduce the power set already here to simplify terminology snapshot fusion. We discuss techniques to find approximations for ϕ in Section 4.

3.2 Terminology Snapshot Fusion

When we have created several separate terminology snapshots, we want to merge them to detect terminology evolution. A term's meaning has evolved if its concept relations have changed from one snapshot to another.

The fusion of two terminology snapshots might be more complicated than just a simple graph merging. For example, we might merge two concepts from the source snapshots to a single concept in the target graph. As part of the fusion process we also need to merge temporal annotations of the edges. When term and concept are equal in both snapshots, the new annotation is just the union of both source annotations. Thus, we represent the concept relations of a term $w \in W$ as set of pairs $(c_i, \{t_{i_1}, \ldots, t_{i_k}\})$. In the left-hand side of Figure 3 we see an example of a graph storing all the term-concept information for term w_1 up to time t_N. To shorten the notation we define τ as a set of time stamps t_i, i.e. $\tau \in \mathcal{P}(T)$ and the pairs can be written as (c_i, τ_i). We note that a concept does not have to be continuously related to a term; instead the respective term meaning/usage can lose popularity and gain it again after some time has passed. Therefore, τ_i is not necessarily a set of consecutive time stamps.

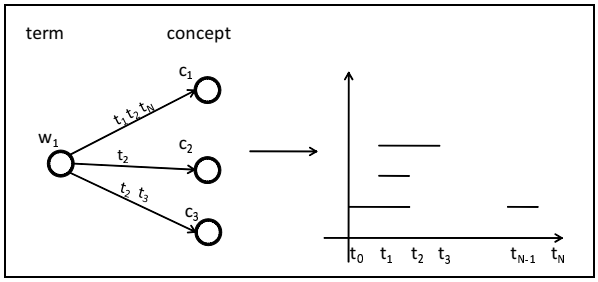

Figure 3: Term-concept relations with temporal annotations.

We introduce the function ψ which fuses two terminology graphs, ψ represents relations between concepts from different snapshots.

$$\psi : (W \times \mathcal{P}(\mathcal{C} \times \tau)) \times (W \times \mathcal{P}(\mathcal{C} \times \tau)) \to (W \times \mathcal{P}(\mathcal{C} \times \tau)) \tag{2}$$

$$((w, \{(c_1, \{t\}), \ldots, (c_i, \{t\})\}), (w, \{(c_j, \tau_j), \ldots, (c_m, \tau_m)\})) \mapsto (w, \{(c_1', \tau_1'), \ldots, (c_n', \tau_n')\})$$

where $c_i, c_j' \in \mathcal{C}, t \in T$ and $\tau_i, \tau_j' \in \tau$ for all i, j. It should be clear that the set of concepts c_i' in the resulting graph of ψ do not necessarily have to be a subset of the concepts $\{c_1, \ldots c_m\}$ from the input graphs.

ψ can be iteratively applied to a term-concept graph from time t_N and the term-concept graph containing all knowledge about a term up to time t_{N-1}.

The graph in the left part of Figure 3 can be interpreted as a timeline of a term where the concepts appear, as it can be seen in the right part of Figure 3. For a term w_1 we follow the timeline and know which concepts were valid at which points in time.

3.3 Mapping Concepts to Terms

The graph resulting from snapshot fusion allows to identify all concepts which have been related to a given term over

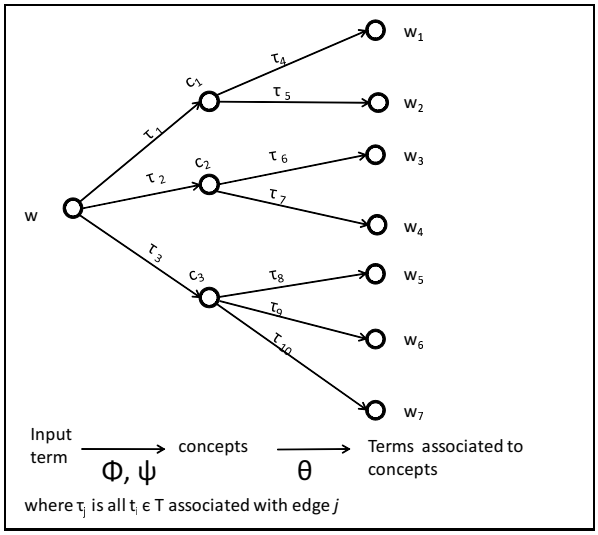

Figure 4: Terminology evolution graph

time. We cannot directly exploit these relations for information retrieval, but need to map the concepts back to terms used to express them. To represent this mapping, we introduce the third (and last) of our functions, θ. For a given concept c, $\theta : C \rightarrow \mathcal{P}(W \times \tau)$ returns the set of terms used to express c, together with time stamp sets which denote when the respective terms were in use.

The characteristics of θ are clearly dependent on how we choose to define the merging operation of the concepts in ψ. For example, if two concepts are merged, the term assignment has to reflect this merge.

3.4 Problem Statement

The steps needed to find terminology evolution are illustrated in Figure 4. We start with an input term w and use ϕ and ψ to map to all concepts ever associated with w. Finally, we map each concept found back to terms using θ.

Based on this model, the task of identifying terminology evolution consists of finding good approximations for the mapping functions ϕ, ψ and θ.

To avoid information overload, we also need strategies which select for a given term only the most relevant related terms, i.e., which select the most relevant paths in the fused terminology graph.

4. RELATED WORK

Temporal aspects in information retrieval come in different flavours like dealing with temporal information within documents, dealing with temporally versioned documents or dealing with temporal evolution of terminologies extracted from documents. An example for dealing with temporal information within documents is the Perseus Digital Library [17]. This digital library system allows accessing and analyzing a number of historic collections in the time, space and language dimensions. Time information is extracted from the documents and can be used for the retrieval.

Little work exists on searching over temporally versioned document collection. An example is the work of Berberich at al. [5] who proposes an extension of the inverted file index to efficiently support temporal search.

Finally, according to our analysis not much work on the problem of terminology evolution has been done. Abecker et al. [1] showed how medical vocabulary evolved in the MEDLINE system. McCray is investigating the evolution of the MESH ontology [2]. In the latter study psychiatric and psychological terms are manually analyzed and their evolution is studied over the past 45 years. For example, there have been major changes in the spectrum disorder terms used for autism.

- 1963: Psychopathology/Schizophrenia
- 1968-1993: Behavior, Mental Process/Thinking, Mental Disorders/Psychotic Disorders and Mental Disorders/Mental Retardation
- 1998-2008: Mental disorder diagnosed in childhood

Also, the concept of sexuality has undergone substantial changes since the 1960s. Previously, between 1963-73, it had been classfied as a sexuality disorder. Today, however, it is classified as sexual behavior.

Terminology evolution can also be observed in other domains. For example in computer science the Faceted DBLP[2] allows to analyze the evolution of used keyword at different times based on the Semantic GrowBag approach [8]. They show that in 2002/03 the term *Sematic Web* subsumes *DAML* whereas in 2003/04 it subsumes *OWL*.

4.1 Relevant technologies

According to our model we have identified three functions ϕ, ψ and θ. In addition terminology extraction is required as a pre-processing step. In this section we discuss relevant technologies for terminology extraction and the approximation of ϕ, i.e. for terminology semantics extraction. ψ and θ heavily depend on how the concepts are represented. Thus technologies for these steps can only be identified and developed once the representation of concepts has been designed.

In the pre-processing step we need to extract the relevant terminology for the domain of the archived collection. Terminology extraction, or glossary extraction, is a subtask of information extraction. Approaches for automatic term extraction make use of linguistic processors, like part of speech tagging, to extract terminological candidates. Generic frameworks like GATE [6] or UIMA [3] allow the flexible composition of extraction and linguistic annotation pipelines and therefore be good starting point to implement our approaches. In [16], a glossary extraction algorithm is presented where candidate glossary items, in the form of noun phrases and verbs, are extracted and then filtered based on confidence level of the pre-modifiers. For example, from *particular vehicle* or *other vehicle* only the *vehicle* term is kept

[2] `http://dblp.l3s.de/`

as a candidate term. Terms that are represented in different forms in the text, e.g. misspelling or abbreviation, are aggregated into single glossary items.

Domain specific terminology can be extracted by comparing the domain corpus with a general language corpus to identify terms that are more significant to the domain. For example, in [11], single word terms are extracted from corpora with different properties. Then statistical comparison of the term frequency is done in order to bring out domain specific terminology. Corpora from the field of telecommunications have been compared with a newspaper corpus. The results have been both automatically validated against a telecommunication terminology database, and manually assessed by telecommunication terminology specialists. The level of precision obtained was reasonably high, varying between 73,0% and 86,1% for the different corpora.

To find ϕ we need to capture the approximate meaning of the extracted terms. One solution for this problem is to automatically discover the senses of words from corpora. Because of the time dependency, it is useful to consider approaches that start from un-annotated text alone. Use of existing dictionaries does not make sense, because they do not reflect the time dependency adequately. There are several approaches for discovering word senses.

In the approach, presented in [9] and [10], a graph $G = (V, E)$ is built using the nouns and their co-occurrences in a collection. Each node $v \in V$ is a noun and there exists an edge $e \in E$ if two nouns "interact". Two nouns are considered to interact if they co-occur in a list; y such as $x_1(, x_2, \ldots and/or\ x_n)$. "Sports such as football, hockey and baseball" would be an example of such a list and in this case *football*, *hockey* and *baseball* become connected nodes in the graph. Clustering methods are then used to cluster the surroundings of a noun. A cluster can be considered a concept of the noun. The relations between the concepts of a noun can reveal Homonymy, Polysemy and Synonymy. A similar but more general approach is presented in [12].

Another approach of word sense discovery focusing on pattern discovery, such as the one presented in [7]. This approach is similar in idea with [10], but it has several advantages, as the approach is more general and the discovered patterns are not hard-coded, part of speech tagging is not required and the graph algorithms used have linear complexity. The approach uses meta-patterns of high-frequency words and content words in order to discover pattern candidates. In the next step they identify the symmetric patterns among the candidates. Their assumption is that two content words are semantically similar if they appear in a symmetric pattern. For identifying the symmetric patterns they build a *single pattern graph* with the content words as nodes, they define a directed edge between two nodes if they appear in pattern in the considered precedence order. Only the symmetric subgraph is then considered, which contains only bidirectional arcs and nodes of the initial graph. In the next phase a graph clique-set method is used for generating initial categories. The approach has been evaluated on both English and Russian corpora, with manual and automatic (using WordNet) assessment.

In [15], a clustering algorithm called Clustering by Committee is presented which automatically discovers word senses by clustering words according to their distributional similarity. Based on top-k similarity between words, the algorithm finds tight clusters of words, called committees and then assigns the rest of the elements to the most similar clusters. For example, the word *heart* is found to belong to the clusters: *kidney, bone marrow, marrow, liver* and *psyche, consciousness, soul, mind*, which can represent its senses. By comparing their results with the WordNet synsets they have measured precision of 63,1% for automatic comparison and 72% for manual comparison. Similarly, [18] have used a lexical context deconvolution algorithm to discover word senses and the syntactic categories of words, and have measured a considerable improvement to the approach in [15].

As a first idea for the computation of ψ the extracted terminology graphs would have to be analyzed. If we represent concepts as groups of words we expect that ideas from [13], where they say that "two objects are similar if they are related to similar objects", can be transferred. Such similarity measured could be used to decide about the merging of concepts in ψ.

5. CONCLUSIONS AND FUTURE WORK

In this paper we discussed the problem of terminology evolution in Web archiving. Adequately dealing with temporal evolution of terminologies is a necessity to ensure that future generations are still able to access past content even if they are not aware of the changes in the meaning of terms. According to our analysis of related work no complete solution for this issue exists. Therefore we started analysing the different types of terminology evolution and introduced a formal model to achieve a better understanding of the problem.

In this paper we present a number of individual relevant technologies, which are natural starting points for the implementation of the evolution detection process. A lot of work has already been done in the field of terminology extraction by using natural language processing techniques which we will also use in our approach. Capturing the senses of terms requires sophisticated methods to analyse term relationships and term clusters for building term-concept graphs. Promising approaches like [7, 9] will be analysed in the next step. Finally, detecting the evolution of term-concept graphs requires the development of new approach inspired by the SimRank algorithm [13].

Overall, our goal for the future is to develop a complete framework that automatically detects the evolution of terminologies and makes it usable for retrieval of web pages and documents in Web archives. The approach will not be limited to web archives but will also be useful for all type of archive and document repository.

6. REFERENCES

[1] A. Abecker and L. Stojanovic. Ontology evolution: Medline case study. In *Proceedings of Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety*, pages 1291–1308, 2005.

[2] Alexa McCray. Taxonomic change as a reflection of progress in a scientific discipline, www.l3s.de/web/upload/talk/mccray-talk.pdf.

[3] Apache. Unstructured information management (uima), 2008. `http://incubator.apache.org/uima/`.

[4] K. Berberich, S. Bedathur, T. Neumann, and G. Weikum. A time machine for text search. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 519–526, Amsterdam, The Netherlands, 2007. ACM.

[5] K. Berberich, S. Bedathur, T. Neumann, and G. Weikum. A time machine for text search. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 519–526, New York, NY, USA, 2007. ACM.

[6] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[7] D. Davidov and A. Rappoport. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 297–304, Sydney, Australia, 2006.

[8] J. Diederich and W. T. Balke. The semantic growbag algorithm: Automatically deriving categorization systems. In *ECDL*, volume 4675 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2007.

[9] B. Dorow. *A Graph Model for Words and their Meanings*. PhD thesis, University of Stuttgart, March 2003.

[10] B. Dorow and D. Widdows. Discovering corpus-specific word senses. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 79–82, Budapest, Hungary, 2003.

[11] P. Drouin. Detection of domain specific terminology using corpora comparison. In *Proceedings of the fourth international Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, 2004.

[12] O. Ferret. Discovering word senses from a network of lexical cooccurrences. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1326, Geneva, Switzerland, 2004.

[13] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, Edmonton, Alberta, Canada, 2002. ACM.

[14] J. Masanès. *Web Archiving*, chapter Web Archiving: Issues and Methods. Springer, 2006.

[15] P. Pantel and D. Lin. Discovering word senses from text. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619, Edmonton, Alberta, Canada, 2002. ACM.

[16] Y. Park, R. J. Byrd, and B. K. Boguraev. Automatic glossary extraction: beyond terminology identification. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Taipei, Taiwan, 2002.

[17] The perseus digital library, 2008. `http://www.perseus.tufts.edu/`.

[18] D. Portnoy and P. Bock. Automatic extraction of the multiple semantic and syntactic categories of words. In *AIAP'07: Proceedings of the 25th IASTED International Multi-Conference*, pages 514–519, Innsbruck, Austria, 2007. ACTA Press.

[19] C. Stokoe. Differentiating homonymy and polysemy in information retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 403–410, Vancouver, British Columbia, Canada, 2005.