

# Applying Optimal Stopping Theory to Improve the Performance of Ontology Refinement Methods

Albert Weichselbraun      Gerhard Wohlgenannt      Arno Scharl  
Vienna University of Economics      webLyzard technology gmbh      MODUL University Vienna  
and Business, Austria      Vienna, Austria      Vienna, Austria  
albert.weichselbraun@wu.ac.at      wohlgenannt@weblyzard.com      arno.scharl@modul.ac.at

## Abstract

*Recent research shows the potential of utilizing data collected through Web 2.0 applications to capture domain evolution. Relying on external data sources, however, often introduces delays due to the time spent retrieving data from these sources. The method introduced in this paper streamlines the data acquisition process by applying optimal stopping theory. An extensive evaluation demonstrates how such an optimization improves the processing speed of an ontology refinement component which uses Delicious to refine ontologies constructed from unstructured textual data while having no significant impact on the quality of the refinement process. Domain experts compare the results retrieved from optimal stopping with data obtained from standardized techniques to assess the effect of optimal stopping on data quality and the created domain ontology.*

## 1. Introduction

Ontologies provide conceptualizations of application domains [10] and facilitate a shared understanding of concepts and relations among different stakeholder groups. The evolution of domain knowledge requires ongoing updates and refinements in the ontology, tasks that are both labor-intensive and costly. Automated ontology learning supports ontology construction and evolution, and helps reduce overall cost and effort.

### 1.1. Ontology Learning

Ontology learning is typically divided into a number of tasks such as the learning of synonyms, concepts, taxonomies, relations and axioms [4]. Many methods in those subfields involve corpus statistics, for example term co-occurrence [18], association rules [19], Latent

Semantic Indexing (LSI) for synonym and concept detection [16] or the application of kernel methods to classify semantic relations [8]. Clustering techniques for taxonomy building employ similarity metrics and frequently exploit the distributional hypothesis [11], which assumes that similar terms or words within a corpus appear in syntactically similar contexts [4]. Ontology refinement is a subprocess of ontology learning which improves existing ontologies, for example, by integration of new knowledge [31].

Building on existing methods to learn ontologies from unstructured data and detect questionable ontological entities, this paper presents a new approach for the dynamic replacement of such questionable entities based on evidence from third-party sources. As they represent the common ground between readers and authors in a given community, many important concepts are never explicitly mentioned in textual data [4]. Structured external sources such as online ontologies (see e.g. d'Aquin et al. [6]) and external social sources in the form of folksonomies [24] help to tackle this issue as they represent rich sources of complementary data in ontology learning.

Cattuto et al. [2] analyze common similarity measures for tags extracted from collaborative tagging systems and investigate implications of those measures for application in ontology learning. The FLOR technique [6] attaches formal semantics to tags which are derived from mappings in online ontologies. Correndo et al. [5] present a method to combine ontology mapping tools with social software techniques which enables users to collaborate on mapping ontologies. Mika [20], Heymann [13], Tang et al. [25] and Schmitz [23] demonstrate ontology learning methods based solely on data retrieved from social sources. In contrast, the approach presented in this paper learns domain-specific ontologies from a domain corpus by extracting relevant terms and relations, and then op-

timizes the process of integrating information from social sources to improve and augment the ontology with external knowledge.

Angeletou et al. [1] note that online ontologies often have a poor coverage on novel scientific terms, multilingual terms and domain specific jargons, whereas folksonomies with their high update frequently tend to reflect the latest vocabulary within domains. Mika [20] discusses the role of semantics emerging from user actions in a community to complement well-established, but slowly evolving ontologies.

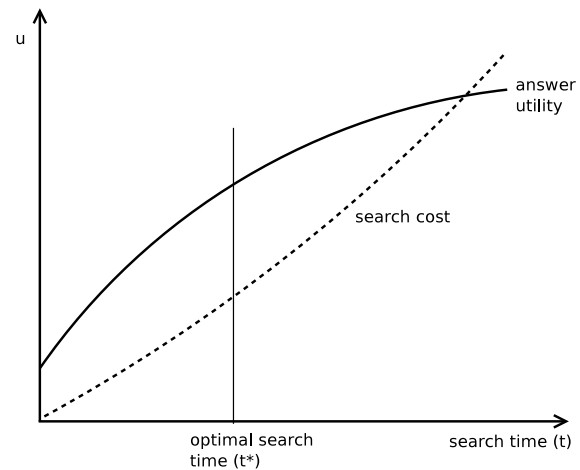
The Web, due to its huge size and heterogeneity, provides an approximation of the real distribution of information of mankind [3]. Although it has a number of shortcomings compared to classical domain corpora such as unknown trustworthiness of sources and inherent noise, many researchers use the Web as repository for information retrieval or knowledge acquisition tasks [22]. Keyword-based search engines such as Google and Yahoo! provide statistics about the information distribution on a significant proportion of the Web. Sanchez and Moreno use Pointwise Mutual Information (PMI), a popular measure applied in Web search context by Turney [27], for the acquisition of labeled relations and terms in ontology learning. They assure domain relatedness of new terms by computing statistical associations between new terms, domain keywords and Web documents. Wong et al. [30] use mutual information between constituents of terms calculated with Web statistics to guide term simplification in the context of ontology learning.

## 1.2. Optimal Stopping

The ontology learning tasks outlined in the preceding section are complex and computationally intensive. Bandwidth considerations and expectations in regards to the system's response time also limit the number of sources that can be included in a feasible manner, as well as the amount of information that can be processed from a single source. Given these limitations, important decisions regarding the optimal allocation of scarce computing and bandwidth resources have to be made either in advance or at run time.

Optimal stopping algorithms such as the Search-Test-Stop (STS) algorithm [28], [29] are effective techniques for making such decisions in distributed network environments. They provide strategies for extracting useful choices from a (usually huge) population without inspecting every single choice. Instead they base the decision on the expected value of the current choice and the statistical properties of the population. Optimal stopping, therefore, optimizes the

trade-off between search cost and the value of the selected choice (see Figure 1). The enormous amount of information available in the Web, combined with new technologies facilitating the exchange and use of external data call for the adaption of utility models and optimal stopping theory in information sciences [9], [28]. As it is no longer possible to analyze all the data available, optimal stopping methods will gain in importance.



**Figure 1. Applying optimal stopping to the information gathering process.**

This development is also reflected by recent research in the fields of (a) *autonomous, and distributed computing*, which applies high level utility functions to reduce the complexity of decision making [15], [26], (b) *information retrieval*, where value driven information retrieval techniques gain in importance [9], [21], [14], [17], and Semantic Web applications, where utility models in conjunction with optimal stopping are applied to decide which data to include into Web portals [28], [29].

## 1.3. Paper Outline

Maedche [19] provides a process oriented view of the ontology learning cycle which consists of the following four phases: import and reuse of existing ontologies if such ontologies are available, extraction and modeling of the target ontology, ontology pruning, and ontology refinement. This paper addresses an interesting sub-problem of the ontology learning process - the refinement of existing ontologies considering the trade-off between the processing time and the improvement of the ontology's quality.

Current work on optimizing Web retrieval only considers isolated Web retrieval tasks such as optimizing

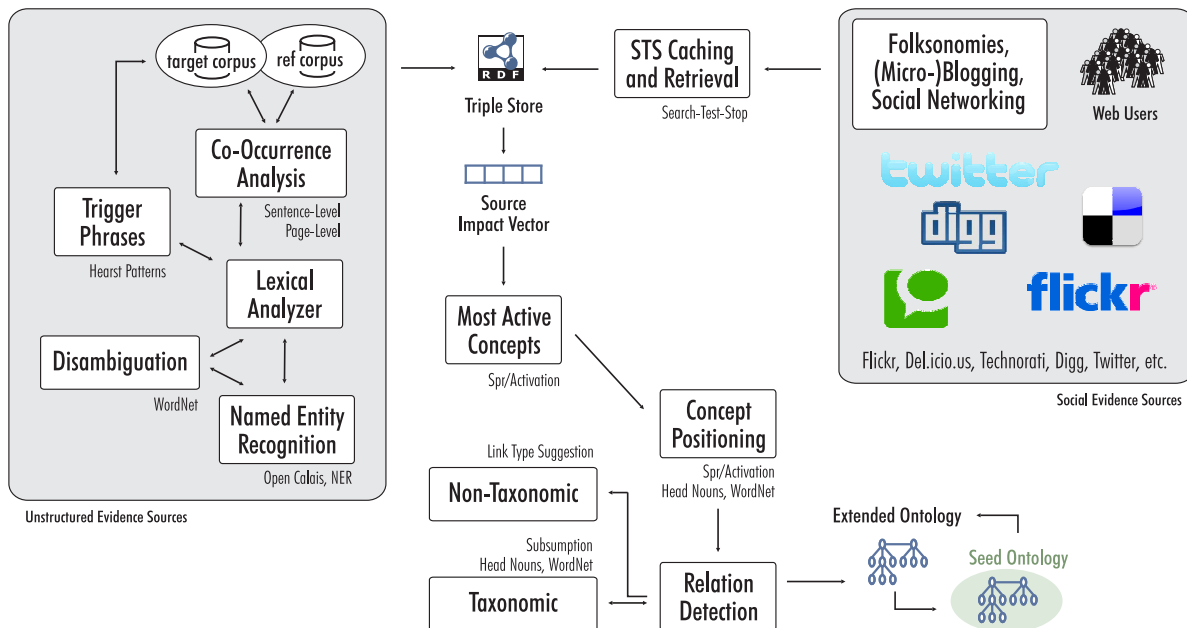


Figure 2. System diagram of the ontology learning architecture.

access to isolated Web services [28] or the more complex issue of applying optimal stopping to geo-tagging [29]. This paper demonstrates how these techniques are applied to a sub-task of ontology learning and may be seen as a first step towards optimizing the whole ontology learning process.

The remainder of this paper is structured as follows: Section 2 presents a method for refining lightweight domain ontologies with terms extracted from social sources and presents an approach for optimizing the refinement process. Section 3 presents an evaluation of this approach. A visualization shows the impact of social evidence sources on the ontology building process and the generated domain ontologies. The paper closes with an outlook and conclusions in Section 4.

## 2. Method

A common approach in state-of-the-art ontology learning is to combine terms extracted from unstructured sources such as Web documents with data retrieved from social evidence sources such as Delicious, Flickr, Technorati and Twitter. Ideally, optimal stopping covers the whole ontology learning process optimizing the access to such sources. Figure 2 illustrates the integration of optimal stopping and caching into ontology learning. Requests to remote resources are piped through an “STS Caching and Retrieval” component which transparently optimizes the data acquisition process.

This paper focuses on demonstrating the usefulness of optimal stopping and caching and, therefore, limits its scope to ontology refinement - a well defined subprocess of ontology learning. We refine ontologies created with the ontology learning component introduced by Liu et al. [18] using data collected from the social bookmarking Web service Delicious and the Yahoo! search engine.

A quality measure for automatically generated domain ontologies (Section 2.1) allows identifying weak terms, which should be replaced during the ontology refinement process (Section 2.2). Two different query strategies suggest candidates to identify weak terms. The evaluation in Section 3 contrasts the quality of the new terms with the processing time of both approaches.

### 2.1. Quality Measures

Dellschaft and Staab [7] distinguish three kinds of ontology evaluation measures covering (i) the *lexical layer*, (ii) the *taxonomy*, and (iii) *non-taxonomic relations*. This work evaluates the domain ontology’s lexical layer by applying an approach suggested by Sanchez and Moreno [22] which combines Web statistics and pointwise mutual information (PMI) to assess the coherence between terms participating in a relation.

A local quality measure ( $\mathcal{M}_l$ ) based on the social source used to refine the ontology (i.e. the PMI of Delicious counts) as well as a global measure ( $\mathcal{M}_g$ ) using Web statistics retrieved from Yahoo! are used

**Table 1. Options considered by the Search-Test-Stop algorithm**

Option (utility)	Description
accept ( $r$ )	The utility obtained by accepting the result based on $x^0$ only. Terms that are so beneficial that the indication provided by $x^0$ is strong enough to justify their acceptance.
drop ( $u^*$ )	The expected utility from replacing the current term with another one. The term is clearly a bad choice, therefore dropping the term and retrieving the next option is expected to yield an utility of $u^*$ (= the average utility of all choices minus search cost).
test ( $T$ )	Retrieving the second indicator $x^1$ and deciding on it will yield the highest utility. The term is a border-case and we require additional data to decide on its fate. Acquiring this data is more beneficial than just accepting or dropping the term.

to judge the quality of terms. The quality measure maps the PMI to three discrete values: *zero* for useless results, *one* for somewhat useful terms and *two* for excellent results.

The correlation between these two indicators has been evaluated based on a list of 314 relations between concept pairs compiled by domain experts and yielded a correlation of 38.5% between  $\mathcal{M}_l$  and  $\mathcal{M}_g$  and 63.4% between  $\mathcal{M}_g$  and the domain experts. The correspondence between the quality measures and the domain experts' evaluations is used to compute their joint probability distribution which is the base for the application of the optimal stopping approach described in Section 2.3.

We apply the global measure  $\mathcal{M}_g$  to all relations to identify low coherence terms in the domain ontologies retrieved from unstructured sources only. The ontology refinement step will replace them with new terms gathered from Delicious.

## 2.2. Ontology Refinement

The ontology refinement step identifies weak relations, as described in Section 2.1, and uses a list of candidate terms from social sources to select a replacement for the weak leaf term that better fits to the ontology's corresponding internal node. We currently compile the list of candidates by retrieving terms which are related to the internal node by using (i) the delicious "Related Tags" sidebar (this approach is limited to monograms such as "climate" only), and (ii) gathering all delicious tags which co-occur with the internal node for n-grams such as "climate change". This paper compares two different strategies to select a term from this list:

- 1) Determine the *best term* from the list by retrieving the local and the global quality measure ( $\mathcal{M}_l$  and  $\mathcal{M}_g$ ) for every candidate term and selecting

the term maximizing the sum of both measures (= brute force).

- 2) Apply the Search-Test-Stop strategy by estimating each term's value based on the local quality measure ( $\mathcal{M}_l$ ). The algorithm stops as soon as a reasonable candidate has been identified. From this point on, continuing searching is expected to produce costs which cannot be offset by a potentially higher utility of another candidate term. The estimate retrieved from the local quality measure might not be decisive. In such cases, the algorithm queries the global quality measure to refine its estimate and to decide whether to continue searching or stop and return the current result. The following section will outline the Search-Test-Stop process in greater detail.

## 2.3. Search-Test-Stop (STS) Algorithm

Search-Test-Stop (STS) is a classic approach from operations research and applied in this work to optimize the trade-off between (i) the time spent searching for and evaluating new terms and, (ii) the utility of the proposed terms. The utility ( $u$ ) corresponds in this paper to the domain expert's assessment of the relation's utility. It, therefore, cannot be determined directly but is estimated by using two indicators. A local quality measure ( $x^0 := \mathcal{M}_l$ ) and a global quality measure ( $x^1 := \mathcal{M}_g$ ). Retrieving these indicators from external sources yields the search cost ( $c^0$ ) and the test cost ( $c^1$ ), which correspond to the response times of the involved services. The indicators provide *estimates* of a term's *true* value which are formalized in the joint probability distribution  $h(x^0, x^1, u)$ . Based on this function we derive (i) the expected utility  $r = E(u|x^0)$  for accepting a term given the indicator  $x^0$  only, (ii)  $u^*$  which is the utility derived from dropping a (bad) term and replacing it with another one, and (iii) the expected utility  $T$  from retrieving  $x^1$  and deciding based on the

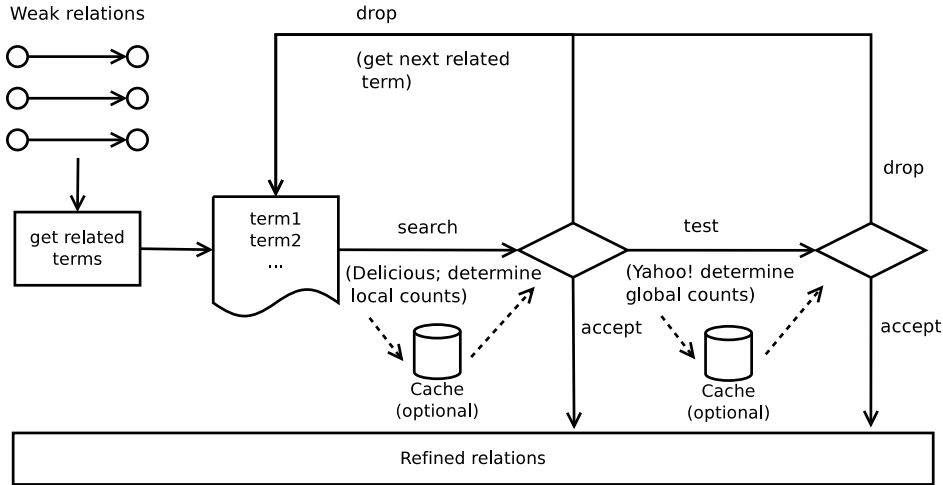


Figure 3. Utility-based ontology refinement.

second indicator. Table 1 summarizes these choices and the corresponding utility measures.

The utility estimates for accepting, dropping and testing the option are computed based on the common probability function  $h(x^0, x^1, u)$ , which is derived from the correlation between the indicators  $x^0, x^1$  and the judgment of domain experts on a list of 314 relations [29]. Using  $h(x^0, x^1, u)$  we compute  $r, u^*$  and  $T$  for every potential candidate based on Hartmann’s solution for the discrete Search-Test-Stop algorithm [12] and then choose the option which corresponds to the estimate yielding the highest utility. Please refer to Weichselbraun [29] for a more detailed description of this process.

Figure 3 illustrates the Search-Test-Stop process for refining the ontology. The refinement component queries social evidence sources to identify candidate terms which could replace low coherence terms. Every candidate is evaluated using a *local* quality measure which is based on the occurrences of the candidate and the seed concept in the social evidence source. The search step, therefore, comprises querying the social evidence source for this data and computing the quality measure  $\mathcal{M}_l$ .

The Search-Test-Stop algorithm [28] decides whether the achieved coherence is high enough to accept or low enough to drop the solution, or whether a testing step, which computes a global quality measure  $\mathcal{M}_g$  based on counts retrieved from the Yahoo! search engine, is necessary. If the answer is accepted, the old term gets replaced by the new term, otherwise the next candidate term from the social evidence source is tested.

## 2.4. Caching

Caching query results is another effective strategy for reducing search and test cost during the ontology learning process. The prototype presented in this paper stores query results for a term’s local (Delicious) and global (Yahoo!) counts using the caching infrastructure provided by the easy Web Retrieval Toolkit ([www.semantic-lab.net/index.php/eWRT](http://www.semantic-lab.net/index.php/eWRT)). Caching transparently redirects search and test queries to the corresponding caches yielding considerably faster response times for cached entries.

## 3. Evaluation

This section outlines the experiments conducted to evaluate the impact of caching and STS on the performance of the ontology learning framework. We initiated the ontology extension process with a small seed ontology of only two relations: *fossil fuels*  $\xrightarrow{\text{relatedTo}}$  *climate change* and *fossil fuels*  $\xrightarrow{\text{relatedTo}}$  *greenhouse gas(es)*. Extending this seed ontology with terms collected from six different domain-specific corpora yields six corresponding domain ontologies for the evaluation process.

Each extension was performed based on unstructured sources (the text corpora described in Section 3.1) based on the method introduced by Liu et al. [18]. Starting with the seed ontology, two iterations of ontology learning extended the ontology by 24 new concepts, which were chosen from the ordered list of automatically generated suggestions.

**Table 2. Uncached performance**

Ontology	Search-Test-Stop			Brute Force		
	$n^0$	$n^1$	total time (s)	$n^0$	$n^1$	total time (s)
April 2009	418	1	928	556	556	3156
May 2009	449	1	1030	623	623	3538
June 2009	558	1	1185	719	719	3920
July 2009	377	1	923	466	466	2611
August 2009	275	1	634	385	385	1879
September 2009	466	1	1037	710	710	3835

**Table 3. Cached performance**

Ontology	Search-Test-Stop			Brute Force		
	$n^0$	$n^1$	total time (s)	$n^0$	$n^1$	total time (s)
April 2009	418	1	390	556	556	1125
May 2009	449	1	380	623	623	1034
June 2009	558	1	299	719	719	632
July 2009	377	1	170	466	466	385
August 2009	251	1	269	385	385	751
September 2009	453	1	490	710	710	1349

### 3.1. Domain Corpora

To create the corpora for the ontology learning, we mirrored 156 news media sites from the News-link.org, Kidon.com and ABYZNewsLinks.com directories. The webLyzard suite of Web mining tools (www.webLyzard.com) crawls those sites in regular intervals, gathering around 200,000 documents per week. Domain detection using a set of regular expressions was used to compile six monthly domain-specific corpora with documents published between April and September 2009. Each of these corpora covers documents of a specific month (e.g. April 2009, May 2009, June 2009, etc.) and has been used to learn the corresponding domain ontology with the approach introduced by Liu et al. [18]. Since the number of documents in each corpus was restricted to 1250, the domain corpora represent just a broad overview of media coverage on the seed concepts in the respective monthly interval.

### 3.2. Identifying and Refining Weak Terms

Computing the PMI between relations of seed terms and terms added in the ontology learning step based on the Yahoo! counts yields the global quality measure ( $\mathcal{M}_g$ ), which provides an estimation of the terms' coherence. Previous experiments suggested that relations with a PMI below 0.75 tend to include unrelated terms. Therefore, an ontology refinement step replaced such terms with more relevant terms retrieved from social sources - i.e. terms related to the seed concept retrieved from Delicious using two decision algorithms, STS and brute force. The following section will compare

these two algorithms with regard to the time spent in refining the ontology as well as the quality of the created ontologies.

### 3.3. Results

Tables 2 and 3 document the performance of the ontology refinement process by contrasting the number of search steps ( $n^0$ ), test steps ( $n^1$ ), and the total time spent by the refinement component. The brute force approach selects the term with the highest expected utility ( $E(u|x^0, x^1) \rightarrow \max$ ), but has to search and test *all* potential candidate terms to ensure a global maximum. STS determines the terms to include in the ontology based on the utility of an option and the statistical properties ( $h(x^0, x^1, u)$ ) of the possible choices. The number of required search and test steps depends on (i) the search cost ( $c^0$ ), (ii) the test cost ( $c^1$ ), (iii) the distribution of the options ( $h(x^0, x^1, u)$ ), and (iv) the first time the algorithm identifies an outstanding option to be accepted.

The evaluation results demonstrate the effectiveness of this method. STS yields performance gains of factor two and higher - even the uncached version of STS outperforms the cached brute force approach in four out of six cases.

Caching considerably reduces the search cost. It, therefore, leads to a higher number of STS search steps (Table 3), since the trade-off between longer search times and better candidate choice has moved towards more (less expensive) searches. Caching does not change the evaluation outcome, in contrast to STS which yields a high-quality but not necessarily the best results. The following evaluation describes the impact of this strategy on the quality of the refined ontologies.



**Table 4. Assessment of the included relations by domain experts along two dimensions: Relevance of the relation to the domain, and domain-independent meaningfulness of the relation.**

Domain Experts	Relation quality (domain specific)		Relation quality (global)	
	STS	brute force	STS	brute force
April 2009	1.700 (6)	1.400 (6)	1.600 (6)	1.267 (6)
May 2009	1.543 (7)	1.514 (7)	1.514 (7)	1.257 (7)
June 2009	1.300 (10)	1.400 (10)	1.040 (10)	1.220 (10)
July 2009	1.514 (7)	1.771 (7)	1.229 (7)	1.543 (7)
August 2009	1.800 (2)	2.000 (2)	1.600 (2)	1.900 (2)
September 2009	1.650 (8)	1.350 (8)	1.500 (8)	1.200 (8)
Sum	1.535 (40)	1.505 (40)	1.360 (40)	1.320 (40)

Figure 4 shows two ontologies (from July 2009) learned with the STS respectively brute force technique integrated into a single graph. Nodes with bold borders depict the concepts from the initial seed ontology. Dotted lines indicate weak relations to candidate terms that were removed with the help of the global quality measure ( $\mathcal{M}_g$ ). Bold edges connect new terms from social sources to existing concepts and thereby replace terms with low coherence; those new terms are distinguished by colored nodes. The shape of nodes distinguish concepts added by STS (rectangular shape) and the brute force approach (octagon shape). If both methods introduced the same concept, the node shape is elliptic.

Table 5 depicts the PMI of relations learned with the help of STS as well as with the brute force approach. As the brute force method determines the global optimum in terms of PMI for all candidate terms, it is only natural that there is a big gap between the two approaches.

**Table 5. Assessment of the included relations using the global quality measure.**

Average PMI	STS	Brute Force
April 2009	0.513 (6)	2.252 (6)
May 2009	0.815 (7)	3.462 (7)
June 2009	0.627 (10)	1.855 (10)
July 2009	0.753 (7)	2.158 (7)
August 2009	1.063 (2)	2.873 (2)
September 2009	0.887 (8)	3.721 (8)
Sum	0.739 (40)	2.673 (40)

STS bases its decision on the statistical properties of the joint probability distribution  $h(x^0, x^1, u)$  which describes the correspondence between the local quality measure ( $x^0 := \mathcal{M}_l$ ), the global quality measure ( $x^1 := \mathcal{M}_g$ ) and the domain experts' assessments of the term's utility ( $u$ ). Consequently, its results yield much lower global PMI's which are even below the threshold of 0.75 used to determine weak terms in the presented experiments. As expected, the differences between the PMI value yielded by the two methods

are highly significant as calculated with a *Welch's t-test* and a *Wilcoxon rank sum test*. Both tests yield *p-values* around  $10^{-15}$  when comparing PMI results for STS and brute force. The values in parenthesis refer to the number of distinct relations per learning method (STS, brute force) in the respective month.

Finally, we compare the utility ( $u$ ) yielded by both approaches. Five domain experts (researchers and Phd students) were asked to evaluate the quality of relations. The evaluation considers two dimensions: (i) the relatedness of the two terms, and (ii) the relevance of the terms (and their relations) to a given target domain ("climate change"). Values of 0/1/2 indicated no/weak/high relatedness or relevance. The level of agreement among domain experts was relatively high, with a standard deviation of 0.36 for dimension (i) and 0.26 for dimension (ii). Table 4 presents the average quality of learned relations according to the domain expert ratings. The overall results from STS and brute force are similar along both dimensions. The human evaluations indicate that STS yields high quality relations for the given domain and input ontology. A *t-test* and a *Wilcoxon rank sum test* confirm the observation that there is no significant difference (p-value of 0.6 for the t-test and 0.8 for the Wilcoxon rank sum test) in domain expert ratings between STS and the brute force method.

## 4. Conclusions

This paper presents the benefits of applying caching and optimal stopping to ontology refinement. An approach originally suggested by Sanchez and Moreno [22] identifies low coherence relations in domain ontologies and replaces them with terms retrieved from Delicious.

The main contributions of this work are (i) introducing the idea of optimal stopping to the field of ontology learning to improve ontology refinement steps which rely on external resources, (ii) designing a framework which identifies low coherence relations in domain



ontologies and refines these relations based on social sources, and (iii) presenting an extensive evaluation based on six different lightweight domain ontologies for elaborating the effects of STS and caching on the performance and quality of ontology learning, showing that STS improves computational performance while having no significant effect on relation quality.

The presented evaluations only use one social source (Delicious), because the main focus of this work has been on demonstrating the benefits of applying optimal stopping and caching to ontology learning. Subsequent work will cover and combine multiple social sources as outlined in the system diagram in Figure 2 (Section 2.1) and address other time intensive aspects of the ontology learning process as well. Future work will also include improved *estimates* of the joint probability distribution  $h$  and complementary coherence measures for identifying weak terms.

## References

- [1] S. Angeletou, M. Sabou, L. Specia, and E. Motta. Bridging the gap between folksonomies and the semantic web: An experience report. In *Workshop: Bridging the Gap between Semantic Web and Web (SemNet 2007)*, volume 2, pages 30–43, 2007.
- [2] C. Cattuto, D. Benz, A. Hotho, and G. Stumme. Semantic Grounding of Tag Relatedness in Social Bookmarking Systems. In *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 615–631. Springer, 2008.
- [3] R. Cilibrasi and P. M. B. Vitanyi. Automatic meaning discovery using google, December 2004.
- [4] P. Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, 2006.
- [5] G. Correndo, H. Alani, and M. Salvadores. Social support for ontological mediation and data integration. *International Journal of Virtual Communities and Social Networking*, 1(3):19–34, 2009.
- [6] M. d’Aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez, and D. Guidi. Toward a new generation of semantic web applications. *IEEE Intelligent Systems*, 23(3):20–28, 2008.
- [7] K. Dellschaft and S. Staab. On How to Perform a Gold Standard Based Evaluation of Ontology Learning. In *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 228–241. Springer, 2006.
- [8] C. Giuliano, A. Lavelli, and L. Romano. Relation extraction and the influence of automatic named-entity recognition. *ACM Transactions on Speech and Language Processing*, 5(1):1–26, 2007.
- [9] J. Grass and S. Zilberstein. A value-driven system for autonomous information gathering. *Journal of Intelligent Information Systems*, 14(1):5–27, March 2000.
- [10] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5-6):907–928, 1995.
- [11] Z. Harris. *Mathematical Structures of Language*. John Wiley & Sons, 1968.
- [12] J. Hartmann. *Wirtschaftliche Alternativensuche mit Informationsbeschaffung unter Unsicherheit*. PhD thesis, Universität Fridericiana Karlsruhe, 1985.
- [13] P. Heymann and H. Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Computer Science Department, April 2006.
- [14] K. Hosanagar. A utility theoretic approach to determining optimal wait times in distributed information retrieval. In *SIGIR ’05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 91–97, New York, NY, USA, 2005. ACM.
- [15] J. O. Kephart and R. Das. Achieving self-management via utility functions. *IEEE Internet Computing*, 11(1):40–48, 2007.
- [16] T. Landauer and S. Dumais. A solution to plato’s problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.
- [17] C. Lim, J. N. Bearden, and J. C. Smith. Sequential search with multiattribute options. *Decision Analysis*, 3(1):3–15, 2006.
- [18] W. Liu, A. Weichselbraun, A. Scharl, and E. Chang. Semi-automatic ontology extension using spreading activation. *Journal of Universal Knowledge Management*, 0(1):50–58, 2005.
- [19] A. Maedche, V. Pekar, and S. Staab. Ontology learning part one - on discovering taxonomic relations from the web. In N. Zhong, J. Liu, and Y. Yao, editors, *Web Intelligence*, pages 301–322. Springer, 2002.
- [20] P. Mika. Ontologies are us: A unified model of social networks and semantics. *Journal of Web Semantics*, 5(1):5–15, 2007.
- [21] A. L. Montgomery, K. Hosanagar, R. Krishnan, and K. B. Clay. Designing a better shopbot. *Management Science*, 50(2):189–206, 2004.
- [22] D. Sánchez and A. Moreno. Learning non-taxonomic relationships from web documents for domain ontology construction. *Data & Knowledge Engineering*, 64(3):600–623, 2008.

- [23] P. Schmitz. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW2006*, Edinburgh, Scotland, 2006.
- [24] L. Specia and E. Motta. Integrating folksonomies with the semantic web. In *The Semantic Web: Research and Applications, 4th European Semantic Web Conference (ESWC-2007)*, volume 4519 of *LNCS*, pages 624–639, Berlin, 2007. Springer.
- [25] J. Tang, H. fung Leung, Q. Luo, D. Chen, and J. Gong. Towards ontology learning from folksonomies. In *IJCAI'09: Proceedings of the 21st international joint conference on Artificial intelligence*, pages 2089–2094, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.
- [26] G. Tesauro, N. K. Jong, R. Das, and M. N. Bennani. On the use of hybrid reinforcement learning for autonomic resource allocation. *Cluster Computing*, 10(3):287–299, 2007.
- [27] P. D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *EMCL '01: Proceedings of the 12th European Conference on Machine Learning*, pages 491–502, London, UK, 2001. Springer-Verlag.
- [28] A. Weichselbraun. Applying optimal stopping for optimizing queries to external semantic web resources. In J. Cordeiro, B. Shishkov, A. Ranchordas, and M. Helfert, editors, *Software and Data Technologies*, volume 47 of *Communications in Computer and Information Science*, pages 105–118. Springer, Berlin-Heidelberg, 2009.
- [29] A. Weichselbraun. Optimizing queries to remote resources. *Journal of Intelligent Information Systems*, 2010. Forthcoming (accepted 2 August 2010).
- [30] W. Wong, W. Liu, and M. Bennamoun. Acquiring semantic relations using the web for constructing lightweight ontologies. In *PAKDD '09: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 266–277, Berlin, Heidelberg, 2009. Springer-Verlag.
- [31] A. J. Yepes, R. B. Llavori, and D. R. Schuhmann. Ontology refinement for improved information retrieval. *Information Processing and Management*, 46(4):426–435, 2010.