# Ontology Evaluation through Text Classification

Yael Netzer, David Gabay, Meni Adler, Yoav Goldberg, and Michael Elhadad

Department of Computer Science
Ben Gurion University of the Negev
POB 653 Be'er Sheva, 84105, Israel
{yaeln,gabayd,adlerm,yoavg,elhadad}@cs.bgu.ac.il

**Abstract.** We present a new method to evaluate a *search ontology*, which relies on mapping ontology instances to textual documents. On the basis of this mapping, we evaluate the adequacy of ontology relations by measuring their classification potential over the textual documents. This data-driven method provides concrete feedback to ontology maintainers and a quantitative estimation of the functional adequacy of the ontology relations towards search experience improvement. We specifically evaluate whether an ontology relation can help a semantic search engine support exploratory search.

We test this ontology evaluation method on an ontology in the Movies domain, that has been acquired semi-automatically from the integration of multiple semi-structured and textual data sources (*e.g.*, IMDb and Wikipedia). We automatically construct a domain corpus from a set of movie instances by crawling the Web for movie reviews (both professional and user reviews). The 1-1 relation between textual documents (reviews) and movie instances in the ontology enables us to translate ontology relations into text classes. We verify that the text classifiers induced by key ontology relations (genre, keywords, actors) achieve high performance and exploit the properties of the learned text classifiers to provide concrete feedback on the ontology.

The proposed ontology evaluation method is general and relies on the possibility to automatically align textual documents to ontology instances.

## 1 Introduction

In this work, we present a new method to evaluate a *search ontology* [1]. The ontology supports a semantic search engine, which enables users to search for movies and songs recommendations in the entertainment domain. Semantic search corresponds to a shift in Information Retrieval (IR) from focus on navigational queries and document ranking to the higher level goals of content extraction, user goal recognition and content aggregation [2][3].

Our search engine operates in a limited domain (entertainment, movies). It relies on an explicit internal ontology of the domain, which captures a structured representation of objects (movies, actors, directors, etc). The ontology is aquired and maintained semi-automatically from semi-structured resources (such

L. Chen et al. (Eds.): APWeb and WAIM 2009, LNCS 5731, pp. 210–221, 2009.

as IMDb and Wikipedia). The ontology supports improved search experience at different stages: content indexing, query interpretation, search result ranking and presentation (faceted search, aggregated search result presentation and search result summarization).

We focus in this paper specifically on evaluating the quality of the ontology as it impacts the search process. As noted by [4], one can distinguish ontology evaluation methods at three levels: structural (measure properties of the ontology viewed as a formal graph), usability (how is the ontology accessed - through API or search tools, versioned, annotated and licensed) and functional (which services does the ontology deliver to applications). The method we present addresses *functional evaluation*, that is, we investigate how one can measure the adequacy of an ontology to support a semantic search engine.

As part of this functional evaluation, we distinguish two forms of information needs expressed by users: fact finding (the user expects to retrieve a precise set of results or to navigate to a specific movie), and exploratory search (the user seeks recommendations for several movies according to non-specific requirements). The ontology provides services to the application for both types of information needs, but in this paper, we focus on support for exploratory search.

The key idea of our evaluation method is that one can evaluate the functional adequacy of an ontology by investigating a corpus of textual documents anchored to the ontology. The textual documents are collected automatically and associated to ontology instances. Hypotheses about the ontology can then be transformed into classification tests on the corpus.

The rest of the paper is organized as follows: we first review previous work in ontology evaluation and ontology-based information retrieval (ObIR). We then present our ontology evaluation method and a set of experiments we ran to evaluate the functional adequacy of our ontology in the entertainment domain. The experiments validate the adequacy of the specific ontology acquired as part of our semantic engine for exploratory search, and provide specific, concrete indications on how to improve the ontology.

## 2 Dimensions of Ontology Evaluation

Evaluation of ontologies is designed and performed according to two main scenarios: assessing the quality of an ontology (by its developers) and ranking ontologies in order to choose the most suitable one for a particular task.

As a general task, evaluation of ontologies is complicated, since ontologies vary in their domain, size, purpose, language and more. Therefore, it is not possible to define a general ontology evaluation paradigm. In addition, the ontology evaluation process depends on the way the ontology was constructed: ontologies may be hand-carved, constructed by scholars or domain experts, or may be the product of an automatic or semi-automatic process. In that case, ontology quality is best measured in terms of cost/profit effectiveness.

Ontology evaluation can focus on one or more of the following dimensions:

- *Functionality (task-based)*: measures how well an ontology serves its purpose as part of a larger application;
- *Usability based*: assesses the pragmatic aspects of the ontology, *i.e.*, metadata and annotation [5];
- *Structural evaluation*: identifies structural properties of the ontology viewed as a graph-like artefact [6].

Among evaluation methods, we distinguish *extrinsic* and *intrinsic* methods. Extrinsic evaluation requires either external information in order to evaluate qualities of the ontology, such as a corpus that represents the domain knowledge (data-driven evaluation), expert opinion, or it requires a particular task which defines the context of the evaluation. Intrinsic evaluation reflects the quality of the ontology as a standalone body of knowledge. Naturally, intrinsic evaluation reflects mostly the structural properties of the ontology.

## 3 Search Ontologies

The usage of an ontology in our current project is motivated by the wish to improve the search experience, *i.e.*, we are interested in evaluating a search ontology as defined in the scope of ObIR (Ontology-based Information Retrieval).

The notion of semantic search refers to search techniques which go beyond the mere appearance of query words in possibly relevant documents, and aims to capture a deeper representation of the searched space and the knowledge embedded in it. Although search is widely used in the Internet, user satisfaction studies indicate that about half of the users complain about irrelevant search results (low precision) or complain about obtaining too many results (see for instance [7]). The usage of an ontology will better address user's expectations, however, it is restricts the scope of a search engine to a specific domain. In our case, we investigate the entertainment domain. For such limited scope search, semantic technology will help the engine find more relevant documents by using links among concepts (*e.g.*, movies with the same actor, similar plot), cluster results along semantic attributes to improve navigation (faceted search), and for conceptual indexing (search for "spy" and get "james bond") [3].

In order to refine the definition of evaluation of a search ontology, we refer first to distinct types of search, which represent different types of information needs (following [8][9]):

- Fact finding: a precise set of results is requested. The amount of retrieved documents is not important (for instance, a specific movie in the entertainment domain). This may correspond to a return visit to a site or a short search session.
- Exploration: the user's need is to obtain a general understanding of the search topic: high precision or recall is not required. For instance, the user explores a movies repository to find interesting movies according to his current mood or similarity with known movies.

- Comprehensive search: the task is to find as many documents as possible on a given topic (high precision and recall), and to organize the resulting set in a synthetic manner. This task is also called “briefing”.

According to these information need distinctions, [9] propose a set of evaluation measures for a search ontology:

- Generic quality evaluation: checks that the ontology is syntactically correct and that it is closely related to the domain.
- Search task fitness: a different measure is applied for each search task. Measures are taken with respect to a cluster of concepts. Fact-finding fitness for a cluster of concepts is a function of the number of instances, properties and data types of all concepts in the cluster. Exploratory search fitness is a function of the number of subclasses, and Comprehensive search fitness is a function of the number of object properties, sub- and super-classes and siblings. (In all cases, the numbers are divided by the number of concepts in the cluster).
- Search enhancement capability measures how useful the ontology is for query expansions, which improve recall and precision. Recall enhancement capability is a function of the number of labels, equivalent classes, intersections and unions of concepts in a given cluster. Precision enhancement capability is a function of the number of all OWL set operations, and of the number data and object properties of concepts in a given cluster.

Such metrics are useful to evaluate ontologies in the same sense that code complexity metrics are useful when developing software. They correspond to what we call intrinsic measures above. These metrics capture the intuition that the search ontology properly supports the operation of a search engine. But these measures do not provide concrete feedback on the functional adequacy of the ontology to the domain. To illustrate the limitations of such intrinsic measures, it is possible to design an ontology to obtain high scores on all metrics with no knowledge of the domain, in a completely artificial manner, by optimizing the distribution of ontology instances across classes. To reuse the software development analogy, code complexity measures are useful to identify “bad code” (functions that are too long for example), but they do not help to assess the correctness or robustness of the code.

Beyond such metrics, we wish to define functional quality criteria for search ontologies. [3] defines the following desirable properties in a search ontology:

- Concept familiarity: the terminology introduced by the ontology is strongly connected to users terms in search queries.
- Document discrimination: the concept granularity in the ontology is compatible with the granularity used in users’ queries. This granularity compatibility allows good grouping of the search results according to the ontology concept hierarchy.
- Query formulation: the depth of the hierarchy in the ontology and the complexity and length of user queries should be compatible.

- Domain volatility: the ontology should be robust in the presence of frequent updates.

This classification of functional quality criteria is conceptually useful, but it does not provide a methodology or concrete tools to evaluate a given ontology. This is the task we address in this paper.

The evaluation methodology we introduce relies on the fact that given an ontology instance (in our domain, a movie), we can automatically retrieve large quantities of textual documents (movie reviews) associated to the instance. On the basis of this automatically acquired textual corpus, we can perform automatic linguistic analysis that determines whether the ontology reflects the information we mine in the texts.

Note that we focus on evaluating the ontology itself and its adequacy to the domain as a search ontology. However, we do not simulate the search process or measure specifically how the ontology affects steps in search operation (such as indexing, query expansion, result set clustering). Accordingly, the evaluation we suggest, although informed by the task (*i.e.*, we specifically evaluate a search ontology), is not a task-based evaluation.

## 4 Experimental Settings: An Ontology for Semantic Search in the Entertainment Domain

We illustrate our ontology evaluation method in the context of the entertainment domain. We first describe quantitative on the experiments we have run. Our project involves the semi-automatic acquisition of an ontology in the movies domain from semi-structured data sources (IMDb, Wikipedia and other similar sources). The objective of our project is to support exploratory search over a set of documents describing movies, actors and related information in the domain.

We first report on **intrinsic evaluation** metrics over the ontology we have been assessing: number of instances, relations, density. Such measures are domain-independent. Interpretation of these measures is eventually task-oriented: we compare the metrics with those established on "high-quality ontologies" in other domains. We use for this purpose the paradigm of OntoQA [10]. Following the definition of a *search ontology*, the ontology is not expected to have a deep hierarchical structure and complex (dense) relations. The basic metrics are illustrated in Table 1. Additional metrics (instance density, relation density) confirm the expectation that the search ontology we assess has a wide and shallow structure.

**Extrinsic evaluation.** considers the two main search types we identified as our target scenario: fact finding and exploratory. In the first scenario, fact-finding search, the user seeks precise results and knows what she should get, the main services expected from the ontology are:

- Produce high precision results and wide coverage for terms used in the queries.
- Provide Named entity recognition functionality to allow fuzzy string matching and identify terminological variations.

**Table 1.** Basic measures of Ontology

| | |
|---|---|
| Classes | 33 |
| Class instances | 351,066 |
| Relations | 27 |
| Relation instances | 19 |
| Movies | 8,446 |
| Persons | 116,770 |

- Identify anchors, *i.e.*, minimal facts that identify a movie (for example, its title, publication year, main actors, main keywords).

For the second scenario, exploratory search, precision and recall cannot be measured since the user does not know apriori what he expects to get. Different criteria have been proposed to assess the quality of an exploratory search system [11]. As mentioned above, we do not attempt a full task-based evaluation, and, therefore, exact quality criteria for exploratory search we identify specific ways through which the ontology can improve the user experience. The services expected from the ontology are:

- Cluster instances by similarity
- Present result-sets using a faceted search GUI to provide efficient browsing and query refinement
- Identify paths of exploration through which movies are identified (period, genre, actors, )

Our task is to assess the adequacy of a specific ontology to provide the services listed above. To address this task, we adopt a corpus-based method: assume we have a corpus of textual documents associated to ontology instances. For example, for each movie instance in our ontology, we have a collection of texts. Our evaluation method translates tests on the ontology into tests on such an aligned textual corpus. We present next two specific tests illustrating this approach – to assess the ontology coverage and its classification adequacy.

## 5 Corpus-Anchored Ontology Evaluation

The first step of our method is to construct a corpus of documents aligned with the ontology instances. In our domain, we construct such a corpus automatically by mining movie reviews from the Web. We collected both professional, edited reviews taken from Robert Ebert's Web site[1] and additional professional and users reviews published in the Metacritic Web site[2] and 13 similar Web sources. The key metadata we collect for each document is a unique identifier indicating to which movie the text is associated. The corpus we constructed for these experiments contains 11,706 reviews (of 3,146 movies). It contains 8.7M words, with an average of 749 words per review.

[1] http://rogerebert.suntimes.com

[2] http://www.metacritic.com

### 5.1 Assessing the Ontology Coverage

To assess the fitness of our ontology to support fact-finding search, we measured the named-entity coverage of the ontology, using the constructed text corpus as reference.

We first gathered a collection of potential named-entity labels in the corpus. In professional reviews, named entities are generally marked in the *html* source. Users' reviews are not edited nor formatted. For such reviews, we relied on Thomson Reuters' OpenCalais[3] named entity recognizer to tag named entities in the corpus.

We then extracted all person names from the textual corpus and searched the labels for each entity in the ontology.

Results show that 74% of the named-entity that appear in professional reviews appear in our ontology. For user reviews (non-edited), the figure is 50%.

The main reasons for mismatches lay in orthography variations (such as accents or transliteration differences), mention of people not related to movie and aliasing or spelling variations (mostly in users reviews). We conclude that the coverage of people's names in ontology is satisfactory; however this test did not take into account variations in names and spelling that are expected.

To investigate terminological variation, we measured the ambiguity level of named-entity labels. By ambiguity, we refer to the possibility that a single name refers to more than one ontology instance. We also measured the level of terminological variation for each ontology instance – that is, given a single ontology instance (*e.g.,* an actor), how many variations of its name are found in the corpus. To identify variations in the text, we used the StringMetrics similarity matching library (http://www.dcs.shef.ac.uk/s̃am/stringmetrics.html). We experimented with the Levenstein, Jaro-Winkler and q-gram similarity measures. For example, using such similarity measures, we could match "Bill Jackson" with "William Jackson".

We have tested coverage on a version of the ontology that included 117,556 instances referring to persons. While taking into account only surnames, we found that 83% of the names are ambiguous. There are 18.57 variations on average for each ontology instance.

This simple exercise indicates how a textual corpus aligned with the ontology and mature language technology (named-entity recognition and flexible string similarity methods) allows us to measure a complex property of the ontology. This evaluation does not only provide a score for the ontology. It also indicates which specific named entities are used in the corpus, how often, which confusions can be expected when disambiguating query terms and how to specifically improve the terminology-related services provided by the ontology.

In the next section, we demonstrate how the more complex task of measuring the clustering adequacy of the ontology can also be assessed using text classification techniques.

[3] http://www.opencalais.com

### 5.2 Assessing the Classification Fitness of an Ontology

As discussed above, the fitness of the ontology to support exploratory search is a function of the number of subclasses. We take this definition a step forward: the number of subclasses is valid if it produces a balanced view of the world domain (represented by the documents) and if the explicit characteristics of the hierarchy can be identified implicitly in the documents.

An ontology induces a hierarchical classification over its elements. Each class (*e.g.*, actor, genre) may be viewed as a dimension for classification of the texts that represent the domain. The ontology provides effective classification services if it meets two criteria:

- The Ontology classification is **useful** if the induced classification is well-balanced, enabling the user explore the dataset in an efficient manner (for exploratory purposes).
- The Ontology classification is **adequate** if the classification induced by the ontology is valid with respect to the domain, which is represented by texts.

Accordingly, we formulate the following hypothesis:

**Hypothesis.** *If* the ontology indicates that some movies are "clustered" according to one of the dimensions, *then* documents associated to these movies should also be found to be associated by a text-classification engine that has been trained on the classification induced by the ontology.

The general procedure we performed to test this hypothesis is the following:

**Step 1:** Choose a dimension to test (we have tested genre, actors and keywords).
**Step 2:** Induce a set of categories (subsets of movies). The subclasses of this dimension and the films instantiated under each subclass defines a clustering of the movies. For example, if we evaluate the "genre" dimension, we cluster movies according to their genre property. In our ontology, this produces about 30 classes of movies (one for each genre value).
**Step 3:** Gather texts (from the reviews corpus, texts that were not used in the acquisition process of the ontology) related to these movies and form a collection ($\text{Text}_{ij}$, $\text{movie}_i$).
**Step 4:** Train a classifier on a subset of the texts ($\text{Text}_{ik}$, $\text{movie}_i$, $\text{category}_i$) where $\text{category}_i$ is the category induced by the ontology.
**Step 5:** Test the trained classifier on withheld data ($\text{Text}_{ij}$, $\text{movie}_i$) and compute accuracy, precision and recall with respect to the category.

**Hypothesis.** Adequate classes yield high accuracy and F-measure on an instance-aligned corpus.

### 5.3 Parameters

There are several reasonable options to perform the text classification task in Step 4 above, with different methods of text representation and with different classifiers.

For text representation, we viewed texts as "bag of words", *i.e.*, as unigrams, and represented each text as a Boolean vector in which each coordinate indicates the existence, or lack of existence, of a string in the text. We tested a few options of preprocessing on the texts and of selecting the features (the strings that we take into account when representing the text): with and without stemming[4] and with and without filtering noise words; selecting features using Mutual Information (MI), or using TF/IDF; and with different numbers of features top 300 or 1000.

Mutual Information-based feature selection is inspired by [12] which shows that this method yields best results on text categorization by topic on a standard News corpus.

The feature selection methods we used are as follows: in TF/IDF, words with the highest values were chosen as features, for the entire corpus. In MI, the features with the highest mutual information associated with the class were chosen (a different set of features is used for every class).

For the classifying task, we used two methods: Support Vector Machines (SVM) (linear and quadratic) and Multinomial Naïve Bayes (MNB) as implemented in the Weka toolkit [13].

### 5.4 Results

We applied the classification procedure to the classification induced by the genre dimension. The classifiers were trained on the reviews corpus. We performed 5-fold cross-validation on the corpus.

The best text representation was established by testing the genre classifier on the task of classification of one class against all.

16 different experimental settings were tested:

- TF/IDF vs. MI.
- Vectors of size 300 vs. 1000 features.
- Stemmed words vs. Raw.
- Noise words filtered vs. no filtering.

For each possibility we tested both SVM and Naïve Bayes as classifiers.

**Classification by Genre.** Genres, according to IMDb.com are defined to be "simply a categorization of certain types of art based upon their style, form, or content. Most movies can easily be described with certain umbrella terms, such as Westerns, dramas, or comedies". The tested ontology includes 23 genre subclasses.

We performed the classification process as described above, and found that the best combination is MI, 1000 features, no stemming, noise filtering, and Naïve Bayes as classifier. The Average F-Measure is 0.41 (all results shown in Table 2). It is possible to explain the failure of the SVM to outperform the Naïve Bayes classifier, due to the imbalanced size of the classes, as shown in [14].

[4] We used the classical Porter Stemmer for the experiment.

**Table 2.** F-Measure of classification engine *One-vs-All*

| **Genre** | **F-Measure** |
|---|---|
| Drama | 0.841 |
| Sport | 0.719 |
| Comedy | 0.709 |
| Thriller | 0.682 |
| Family | 0.626 |
| Adventure | 0.625 |
| Action | 0.616 |
| Documentary | 0.613 |
| Sci-Fi | 0.551 |
| Horror | 0.540 |
| Animation | 0.539 |
| Fantasy | 0.533 |
| Music | 0.500 |
| Crime | 0.490 |
| Romance | 0.462 |
| Western | 0.431 |
| Mystery | 0.352 |
| History | 0.289 |
| Musical | 0.274 |
| War | 0.257 |
| Short | 0.239 |
| Biography | 0.231 |
| Adult | 0.198 |

**Table 3.** Pair Classification of Genres

| **Pair** | **F-Measure** | **Accuracy** |
|---|---|---|
| Drama - Western | 0.997 | 0.994 |
| Drama - Musical | 0.996 | 0.991 |
| Thriller - Musical | 0.986 | 0.972 |
| Action - Western | 0.979 | 0.960 |
| Thriller - Western | 0.977 | 0.956 |
| Action - War | 0.935 | 0.894 |
| War - Action | 0.438 | 0.809 |
| Adult - Romance | 0.367 | 0.773 |
| Biography - Documentary | 0.358 | 0.739 |
| History - Short | 0.343 | 0.821 |
| Adult - Short | 0.287 | 0.702 |
| Biography - Drama | 0.172 | 0.903 |

The results indicate that some genres are very well defined (drama, sport, comedy), while others cannot be recovered by analyzing the text of the reviews (musical, short, biography, adult).[5] While these figures provide a first assessment of the quality of each genre category, pair-wise classification provides finer-grained tests of the level to which pairs of genres can be distinguished. A subset of the results showing best and worst cases is shown in Table 3. We report both F-measure and Accuracy for these tests.

The average error rate for pairwise classification is 16.2%; it varies significantly between genre pairs, and therefore can indicate a weak category or classes which are harder to differentiate.

[5] Specifically, the genres of music and musical are derived from the IMDb genres and are apparently confusing.

For comparison, we have tested a Baseline classifier which is not related to the Ontology under test in any way. This was done by creating 25 random classes of 1,000 movies. We performed the same classification procedure. The results showed average F-measure lower than 0.16 (as opposed to 0.41 overall for the ontology-based classifiers, and over 0.70 when we filter out low-quality genres) and extremely low accuracy (less than 0.1). This indicates that the corpus-anchored ontology evaluation method does not capture random patterns of text classification.

Note that the pair-wise classifiers are not symmetric: this is because there can be overlap between two categories. For example, a movie can belong both to the genres of action and drama. In our experiment, when we test the pair drama-action, we learn a binary classifier that responds "true" for texts classified as drama, and "no" for all the rest. This classifier is only trained over documents associated to movies that are tagged as either drama or action (all other texts are ignored). If a movie is tagged as both drama and action, it will be classified as "true" for the drama-action classifier as well as for the action-drama classifier. This asymmetry provides an indication that one genre may be included in another.

## 6 Conclusion and Future Work

We have presented a concrete ontology evaluation method based on the usage of a corpus of textual documents aligned with ontology instances. We have demonstrated how to operate such evaluation in the case of an ontology in the entertainment domain used to improve a semantic search engine.

We have first constructed an ontology-aligned textual corpus by developing a Web crawler of movie reviews.

Our first experiment measures the adequacy of the ontology to support fact-finding search. We have found specifically that our ontology has wide coverage but lacks support for ambiguity resolution and terminological variation handling. We use human-language technology to translate hypothesis on the ontology coverage into measures of properties of the textual.

Our second experiment measures the adequacy of the ontology to support exploratory search. We have formulated hypotheses that capture the quality criteria of an exploratory search system, and tested these hypotheses on our ontology-aligned textual corpus. Specifically when testing the classification adequacy of our ontology along the "genre" dimension, we found that most of the genres in the ontology induce high-quality text classifiers - but some, such as sport and music) do not induce appropriate classifiers. This method provides specific feedback to the ontology maintainer.

Our tests support the claim that classification as a method for evaluation is adequate.

## Acknowledgments

This research was supported by Deutsche Telekom Laboratories at Ben-Gurion University of the Negev. We wish to thank Jon Atle Gulla and Jin Liu for fruitful collaboration.

## References

1. Burkhardt, F., Gulla, J.A., Liu, J., Weiss, C., Zhou, J.: Semi automatic ontology engineering in business applications. In: Proceedings of the 3rd International AST Workshop – Applications of Semantic Technologies. LNI, vol. 134, pp. 688–693 (2008)
2. Baeza-Yates, R., Ciaramita, M., Mika, P., Zaragoza, H.: Towards semantic search. In: Kapetanios, E., Sugumaran, V., Spiliopoulou, M. (eds.) NLDB 2008. LNCS, vol. 5039, pp. 4–11. Springer, Heidelberg (2008)
3. Gulla, J.A., Borch, H.O., Ingvaldsen, J.E.: Ontology learning for search applications. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 1050–1062. Springer, Heidelberg (2007)
4. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modelling ontology evaluation and validation. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 140–154. Springer, Heidelberg (2006)
5. Gomez-Perez, A.: Evaluation of ontologies. International Journal of Intelligent Systems 16, 391–409 (2001)
6. Alani, H., Brewster, C.: Ontology ranking based on the analysis of concept sructures. In: Proceedings of the 3rd International Conference on Knowledge Capture (K-Cap), Banff, Canada, pp. 51–58 (2005)
7. JupiterResearch: Search technology buyerś guide. Technical report, IBM Content Discovery (2006), `ftp://ftp.software.ibm.com/software/data/cmgr/pdf/searchbuyersguide.pdf`
8. Aula, A.: Query formulation in web information search. In: Proceedings of IADIS International Conference WWW/Internet, pp. 403–410 (2003)
9. Strasunskas, D., Tomassen, S.L.: Empirical insights on a value of ontology quality in ontology-driven web search. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part II. LNCS, vol. 5332, pp. 1319–1337. Springer, Heidelberg (2008)
10. Tartir, S., Arpinar, I., Moore, M., Sheth, A., Aleman-Meza, B.: OntoQA: Metric-based ontology quality analysis. In: Proceedings of Workshop on Knowledge Acquisition, Autonomous, Semantically Heterogeneous Data and Knowledge Sources, pp. 45–53 (2005)
11. White, R.W., Muresan, G., Marchionini, G. (eds.): ACM SIGIR Workshop on Evaluating Exploratory Search Systems, Seattle (2006)
12. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: Proceedings of the Seventh international Conference on Information and Knowledge Management, Bethesda, Maryland, pp. 2–7 (1998)
13. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
14. Akbani, R., Kwek, S.S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004)