Automatic Evaluation of Search Ontologies in the Entertainment Domain using Natural Language Processing

Michael Elhadad, David Gabay, and Yael Netzer
(elhadad—gabayd—yaeln)@cs.bgu.ac.il
Department of Computer Science

Ben Gurion University, Israel

Abstract. Information Retrieval (IR) research has recently started addressing the information need of *e*xploratory search, where the searcher may be unfamiliar with the domain or not have decided what is the goal of his query. A popular approach to support exploratory search is the usage of *faceted search*. The implementation of faceted search requires that documents be annotated by metadata in the form of attributes and hierarchical categories. In many applications, the metadata is maintained manually, in the form of a search ontology. Recent work has also investigated methods to automatically acquire such metadata from sample documents. In this work, we propose a new method to automatically evaluate the quality of such a search ontology.

Our method relies on mapping ontology individuals to textual documents. On the basis of this mapping, we evaluate the adequacy of the ontology by translating ontology properties into properties over the textual corpora, which can be empirically tested using natural language processing techniques. This data-driven method provides concrete feedback to ontology maintainers and quantitative estimation of the functional adequacy of the ontology towards search experience improvement.

We test this ontology evaluation method on an ontology in the Movies domain, that has been acquired 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 individuals by crawling the Web for movie reviews. The 1-1 relation between textual documents (reviews) and movie individuals in the ontology enables us to test the ontology terminological coverage, its classification potential and its capability to capture the topic of movies. Since the proposed ontology evaluation method relies only on the possibility to align textual documents to ontology individuals, it is domain independent and can be applied to various, well-documented domains.

1 Introduction

Faceted search is a popular interaction method, which allows users to navigate through complex unfamiliar data [1, 2]. Faceted search has been widely

used in recent e-commerce sites and spread by software providers such as Endeca. Faceted search supports the information need of *exploratory search* [3]. Exploratory search corresponds to a shift in IR from focus on navigational queries and document ranking to the higher level goals of content extraction, user goal recognition and content aggregation [4].

When using a faceted search interface, a user first submits a general query, and then navigates through several facet hierarchies that describe the result set. By navigating through the facet hierarchies, the user interactively refines his query and discovers further facets that may guide him through the document repository and help him discover relations he did not know when first issuing his query. Typically, faceted search applications show possible refinements to the current query together with the number of search results that the refinement would bring in focus. These counts provide helpful quantitative feedback to the user and guide him when exploring the data.

The effectiveness of this interaction strategy relies heavily on the quality of the metadata associated with the documents: which attributes (facets) describe the documents, and the hierarchy of values associated to each facet. In many applications such as e-commerce Web sites, information architects carefully prepare this metadata in the form of a search ontology [2, 5, 6]. Some researchers have also investigated automatic and semi-automatic methods to construct such a search ontology by mining information in the document repository and related resources [7–9].

Both when manually maintaining the search ontology or acquiring it automatically, there is a pressing need to evaluate its quality. In this work, we present a new method to evaluate a *search ontology* [10]. We tested our method specifically in the entertainment domain: a semantic search engine enables users to search for movies and songs recommendations. The search engine relies on an explicit internal ontology of the domain, which captures a structured representation of objects (movies, actors, directors...). The ontology is acquired and maintained semi-automatically from semi-structured resources (such 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 [11], 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 individuals. They do *not* correspond to the document repository covered by the search engine, but are only used for ontology evaluation. Hypotheses about the ontology can then be transformed into classification tests on this test corpus.

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 illustrating this method 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

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 constructed manually, by scholars or domain experts, or may be the product of an automatic or semi-automatic process. In most applications, ontology quality is best measured in terms of cost/profit effectiveness.

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

- *Structural evaluation*: identify structural properties of the ontology viewed as a graph-like artifact [12].
- *Usability*: assess the pragmatic aspects of the ontology, *i.e.*, metadata and annotation [13];
- *Functionality*: measure how well an ontology serves its purpose as part of a larger application;

It is also useful to distinguish between *extrinsic* and *intrinsic* evaluation methods. Extrinsic evaluation requires either external information in order to evaluate qualities of the ontology, such as expert opinion, a corpus that represents the domain knowledge (data-driven evaluation), or 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.

The method we propose is extrinsic. We assess the adequacy of an ontology to support exploratory search (in the form of faceted search) by constructing a set of textual documents derived from the ontology and testing hypotheses on this

dataset. This methodology is data-driven and automatically provides concrete feedback to the ontology maintainer. In our approach, the set of naturally occurring documents on which we perform the evaluation provides an approximation of expert opinion.

3 Search Ontologies

The usage of an ontology in our 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).

Semantic search refers to search techniques that go beyond the mere appearance of query words in possibly relevant documents, and aims at capturing 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 general user insatisfaction about irrelevant search results (low precision) or about obtaining too many results. The usage of an ontology can better support users expectations, at the cost of restricting 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 helps 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)[14], and for conceptual indexing (search for "spy" and get "james bond") [8].

In the rest of the paper, we refer to the elements in an ontology using the following terminology: the ontology describes *concepts* using a description language (*e.g.,* OWL [15]). Concepts denote a *class of individuals* and are related to each other through *relations*. One of the most important relation is that of *specialization*, indicating that a concept *Child* is a subset of a more general concept *Parent*. When the ontology is populated, each concept is "filled" by specific individuals. For example, the concept *Movies* would be filled by thousands of individuals that denote each movie instance in the ontology. Each individual is described by its concepts (the classes to which it belongs) a set of data properties (*e.g.,* release date), and relations to which it belongs (*e.g.,* a movie individual stands in relation with individuals of the actor class). The names of the individuals, of the classes and of the relations determine the *terms* defined by the ontology.

In order to define the evaluation of a search ontology, we refer first to distinct types of search, which represent different types of information needs (following [16][17]):

- **Fact finding:** a precise set of results is requested. The number of retrieved documents is expected to be small (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 [3].

According to this information needs distinction, Strasunskas and Tomassen [17] 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. Fact-finding fitness for a cluster of concepts is a function of the number of individuals, properties and data types of all concepts in the cluster. *Exploratory search fitness* is a function of the number and distribution of subclasses.
– 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 equivalent classes. Precision enhancement capability is a function of the number of concept properties.

Such metrics are useful to evaluate ontologies in the same sense that code complexity metrics are useful when developing software or readability index when assessing the quality of text. 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 individuals 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. Gulla et al. [8] define the following desirable properties in a search ontology:

– Concept familiarity: the terminology introduced by the ontology is strongly connected to user 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, specifically for exploratory search.

The evaluation methodology we introduce relies on the fact that given an ontology individual (in our domain, a movie), we can automatically retrieve large quantities of textual documents (in our case, movie reviews) associated to the individual. 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 (*i.e.*, we do not evaluate the ontology on a search benchmark).

4 Experimental Setting: An Ontology for Semantic Search in the Entertainment Domain and a Test Corpus

We illustrate our ontology evaluation method in the context of the entertainment domain. We first describe the experimental setup. The objective of our project is to support exploratory search over a set of documents describing movies, actors and related information in the domain. The ontology we evaluate is automatically acquired from semi-structured data sources (IMDb, Wikipedia and other similar sources).

Table 1 shows the size of the ontology we use in our experiments. As is appropriate for a *search ontology*, the ontology is wide and shallow.

Table 1. Size of the Movies Ontology

Classes	33
Class individuals	351,066
Relations	27
Movies	8,446
Persons	116,770

Our ontology evaluation method includes the following steps: first, we compiled a corpus of texts from the domain, distinct from the documents used for the acquisition of the ontology. We then used standard Natural Language Processing (NLP) techniques to evaluate the ontology by testing various hypotheses on the collected corpus. We report on the following three experiments:

– Measure coverage and term alignment: we attempt to test the adequacy of the ontology with respect to *concept familiarity*, (cf. Section 3). This *Coverage Experiment* is discussed in Section 5.

- Measure classification fitness on movie genres: we attempt to test the potential of the ontology to properly organize movies into genres. Genre (*e.g.,* comedy, drama) is a critical metadata attribute of movies. This *Classification Experiment* is described in Section 6.
- Measure topic identification fitness: we assess the capacity of the ontology to capture the notion of "topic" of movies. Topics describe what the movie is about and are distinct from its genre. They are most often described by keywords. This *Topic Experiment* is described in Section 7.

Each of these experiments exploits a different technique of NLP. For the coverage and term alignment experiment, we use fuzzy string matching techniques and Named Entity Recognition (NER). For the classification experiment, we use text classification and for the topic experiment, the LDA topic modeling method [18].

For all experiments, we use the same test corpus. We construct this corpus in such a way that documents are aligned to ontology individuals. In our domain, we contruct the 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 user 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.

5 Evaluating the Ontology Coverage of Query Terms

5.1 Objective

Consider a fact-finding search scenario: the user seeks precise results and knows what results she should get. The main services expected from the search ontology to support this scenario are:

- Produce high precision results and wide coverage for terms used in the queries
- Provide entity recognition functionality to allow fuzzy string matching and identify terminological variations.
- Identify anchors, *i.e.*, minimal facts that identify a movie (for example, its title, publication year, main actors, main keywords).

Given a populated ontology, we want to assure that the individuals in the ontology match the entities in the actual domain (that is, the ontology describes the right set of individuals). We also want to verify that the way the ontology individuals are retrieved corresponds to the terminology used in actual search queries (that is, the ontology uses the correct terms to refer to individuals). Our experiment assesses these two dimensions by testing hypotheses on the test corpus.

[1] http://rogerebert.suntimes.com
[2] http://www.metacritic.com

5.2 Hypothesis

We want to test whether the terminology used in the ontology to refer to entities (movies, actors, directors) corresponds to the terminology used in queries. We do not have access to a search query log, nor to manually annotated data that would match terms in a query log to ontology individuals. Instead, we use our test corpus as a proxy: we consider that naturally occurring text about movies is similar (as far as references to movies and actors is concerned) to queries users would submit.

The task we address is unsupervised: the corpus is not annotated manually (this would be an expensive operation). We want to assess the extent to which a named entity in text can be mapped to a term in the ontology.

The hypothesis we formulate is the following: If the ontology has good term coverage then:

1. Named entities found in text will be found in the ontology (coverage is high).
2. The mapping from a named entity found in text to an individual in the ontology is accurate (there is no ambiguity).

5.3 Assessing the Ontology Coverage

To assess the ontology coverage, we measured the overlap between named-entities that appear in the corpus and the terms that appear in the ontology.

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. User reviews are not edited nor formatted. For such reviews, we relied on the OpenCalais [3] named entity recognizer (NER) to tag named entities in the corpus. This NER system recognizes multi-word expressions that refer to proper names of people, organizations or products.

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-entities that appear in professional reviews also appear as terms 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 in the reviews and aliasing or spelling variations (mostly in user reviews).

We conclude that the coverage of people entities in the ontology is satisfactory. However, it is not certain that a search for these entities will find them or will find the intended individual in the ontology. This fuzziness is caused by term variation (as observed especially in user reviews) and term ambiguity.

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

5.4 Assessing the Ontology Terminological Precision

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 individual. By variation, we refer to the opposite case, where one ontology individual can be described by various terms in text.

We measured the level of terminological variation for each ontology individual – that is, given a single ontology individual (*e.g.,* an actor), how many variations of its name are found in the corpus (Bilenko and Mooney [19] used a similar method in a different setting). To identify variations in the text, we used the StringMetrics similarity matching library[4]. We experimented with the Levenstein, Jaro-Winkler and q-gram similarity measures. For example, using such similarity measures, we could match "Bill Jackson" (a name often used in blogs to informally refer to the actor) with "William Jackson" (the name under which the actor is described in the ontology). Such flexibility in aligning query terms with ontology terms increases the search system recall – but it also introduces risks of lost precision. Loss in precision is introduced when two distinct individuals in the ontology can be named by the same term. For example, if there is an actor named "Bill Jonson" and another one named "William Johnson" in the ontology – a fuzzy string matching would confuse the two actors.

In order to measure the practical impact of the factor of name variability and ambiguity, we extracted information from the corpus of movie reviews we collected. To perform this analysis, we first developed a named entity recognizer (NER) specialized to the Movies domain. (The OpenCalais NER we used above properly tags person names, but it cannot distiguish actors and directors and it cannot identify movie names). To this end, we manually tagged a corpus of 200 movie reviews from the Ebert corpus, to indicate the occurrence of movie names and actor names. We then applied the YAMCHA [20] package to train an automatic NER system on our corpus. YAMCHA uses a support vector machine (SVM) classifier to recognize named entities in text based on features describing each word. We used two different sets of features to train the system: whether words start with capital letters (a strong indication that the word is a proper name in English) and whether the name occurs in a gazette (a list of proper names manually compiled). We also used contextual features (properties of the words around the word to be classified). The main challenges a statistically trained NER system addresses are (1) to identify a sequence of several words as a single named expression (for example, "Bill Gates 3rd") and (2) to recognize through generalization that words that do not appear in a pre-defined gazette of names are similar in their distribution to known names so that terms that have not been observed before as proper names are properly recognized as such. A NER system must also use contextual information to avoid tagging a word that does appear in the gazette but is not used as a proper name (for example "Bill paid the bill"). Finally, a NER system must distinguish between proper names referring to actors or to movies.

[4] http://www.dcs.shef.ac.uk/sam/stringmetrics.html

Table 2. Performance of the Named Entity Recognizer on the Movies Domain

	precision	recall	F
Movie exact match	91.56	92.43	91.87
Movie boundary match	91.80	92.66	92.10
Person exact match	97.68	93.76	95.67
Person boundary match	97.84	93.92	95.83
Average accuracy	99.32		
Average boundary accuracy	99.34		

Results Table 2 summarizes the results of the trained NER system. The system on average is capable of properly identifying 92% of the movies named in the reviews and 94% of the persons.

Let us now consider a given text as a search query. If we apply our NER system on the text of the query, we will properly tag named movies or actors. The issue we now address is: how successful will we be in aligning a named entity in the query with the corresponding individual in the ontology?

For a version of the ontology that includes 117,556 individuals referring to persons, taking into account surnames only, we found that 83% of the names may refer to more than one instance of the ontology. We also found that over 18 name variations on average for each ontology instance actually occurred in the corpus.

5.5 Ontology Terminological Coverage: Conclusions

In conclusion, while the coverage of the ontology originally looked promising, we found that because of the issue of name variability and ambiguity, aligning a query with ontology individuals remains an acute challenge. In this experiment, we used the corpus of reviews as a representative set of search queries. Note that we do not claim that reviews are similar to actual search queries. We only claim that the way movies and actors are named in naturally occurring reviews is a good indication of how they will be named in search queries.

This 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 improve the terminology-related services provided by the ontology. The dataset we have collected can then be used to specifically tune the specific method to be used to align named entities with ontology individuals.

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

6 Evaluation of the Ontology Classification Fitness

6.1 Objective

Consider an exploratory search scenario: precision and recall cannot be measured since the user does not know *a priori* what he expects to get. Different criteria have been proposed to assess the quality of an exploratory search system [21]. We do not attempt a full task-based evaluation (that would be expensive). Instead, we identify quality criteria for exploratory search that enable specific ways through which the ontology can improve the user experience. The services expected from the ontology are:

- Cluster individuals by similarity to address the need to present large result-sets.
- 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, ...) to structure a sequence of queries.

Our task is to assess the adequacy of our ontology to provide the services listed above. Again, we adopt a corpus-based method: our evaluation method translates tests on the ontology into tests on our aligned test corpus.

6.2 Hypothesis

The fitness of the ontology to support exploratory search is a function of the number of classes which organize the range of individuals. We take this definition a step forward: the structure of ontology classes is valid if it produces a balanced view of the world domain (represented by the documents) and if the explicit characteristics of the structure can be identified implicitly in the documents.

The ontology induces a classification over its individuals. 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 to 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: 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.

6.3 Classification Experiment

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

Step 1: Choose a dimension to test (we report here on genres).
Step 2: Induce a set of categories (subsets of movies). The subclasses of this dimension and the movies instantiated under each subclass define 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 subsets 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_{ij}, movie_i).
Step 4: Train a classifier on a subset ($\text{Text}_{ik} \subset \text{Text}_{ij}$) of the texts ($\text{Text}_{ik}$, movie_i, category_i) where category_i is the category induced by the ontology.
Step 5: Test the trained classifier on withheld data (Text_{ij}, movie_i) and compute accuracy, precision and recall with respect to the category.

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 "bags of words", *i.e.*, as unigrams, and represented each text as a boolean vector in which each coordinate indicates the presence or absence of a string in the text. We tested several 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[5] 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.

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). Mutual Information-based feature selection is inspired by [22], which shows that this method yields best results on text categorization by topic on a standard News corpus.

For the classifying task, we used two methods: Support Vector Machines (SVM) (linear and quadratic) and Multinomial Naive Bayes (MNB) as implemented in the Weka toolkit [23].

6.4 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.

[5] We used the classical Porter Stemmer for the experiment

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 Naive 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, 300 features, no stemming, noise filtering, and Naive Bayes as classifier.

There are several ways to measure the performance of a binary classifier. A common measure in natural language processing tasks is the F-measure, which is defined by:

$$\frac{2TP}{2TP + FP + FN}$$

Where TP is the number of elements in the 'positive' class that were correctly classified, FP is the number of the elements in the 'negative' class, falsely classified as positive, TN is the number of correctly classified negative elements and FN the number of elements in the positive class, classified as negative.

F-measure takes values between 0 (always mistaken) and 1 (always correct) and it combines in a single metric the desire to obtain high precision while maintaining high recall. For our task, we found Matthews correlation coefficient (MCC) more suitable. MCC is given by:

$$\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}}$$

The values of MCC range between -1 (always wrong) and 1 (always correct). The Average F-Measure is 0.49, and the average MCC is 0.43 (all results shown in Table 3). The average F-measure and MCC are reasonably high. For comparison, we have tested a Baseline classifier: 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 and extremely low accuracy.

This indicates that the corpus-anchored ontology evaluation method does not capture random patterns of text classification. In other words, knowledge extracted from the ontology helped us classify text previously unseen in the domain covered by the ontology. This overall high performance on the task of

Table 3. MCC, F-Measure and error rates of classification engine One-vs-All, sorted by MCC

Genre	**%Error**	**F-Measure**	**MCC**
Sport	2.49	0.72	0.71
Family	5.59	0.63	0.60
Documentary	1.01	0.61	0.56
Adventure	11.88	0.63	0.56
Comedy	21.19	0.71	0.56
Thriller	20.06	0.68	0.54
Action	14.00	0.62	0.53
Sci-Fi	7.94	0.55	0.52
Animation	5.68	0.54	0.52
Horror	8.88	0.54	0.51
Fantasy	7.47	0.53	0.50
Music	3.85	0.50	0.48
Drama	22.18	0.84	0.47
Western	2.35	0.43	0.46
Crime	16.96	0.49	0.39
Romance	23.17	0.46	0.31
Musical	1.73	0.27	0.29
Mystery	12.50	0.35	0.28
History	5.30	0.29	0.28
War	5.97	0.26	0.23
Biography	6.25	0.23	0.22
Adult	6.43	0.19	0.18
Short	20.91	0.24	0.18

text classification validates our basic hypothesis that we can perform tests on the ontology by computing properties of the corresponding test corpus.

What we are now interested in investigating is: how can we exploit the details of the quantitative analysis of the trained text classifiers to learn specific properties of the ontology. In a typical classification scenario, the classifier *itself* is the object of evaluation (typically, several classifiers are compared on the same dataset), and hence the correlation between class size and F-measure is not an issue; in our case, it is the classes, not the classifier, that we want to evaluate. The advantage of MCC over F-measure comes from the fact that unlike F-measure, it is not affected by the "positive" class size. A random binary classifier, that does not consider the content but only the observed probability P of the positive class in the training set, is expected to yield an F-score of P. Its MCC is expected to be 0. Another point in favour of MCC is that it is symmetrical and thus more suitable to the case of classifying pairs of genres, in which there is no natural choice of the "positive" class.

The results indicate that some genres are very well defined (sport, family, documentary), while others cannot be recovered by analyzing the text of the re-

views (history, war, biography, adult, short).[6] While these figures provide a first assessment of the quality of each genre category, pairwise classification provides finer-grained tests of the level to which pairs of genres can be distinguished. In pairwise classification, we first classify each pair of genres (*e.g.,* drama vs. history) and then calculate the average accuracy measure over all pairs. Pairwise classification indicates genres pairs that are similar (with low accuracy measures). A subset of the results showing best and worst cases is shown in Table 4. We report MCC and error rate for these tests.

Table 4. Pairwise Classification of Genres

Pair	**% Error**	**MCC**
Sport - Fantasy	4.76	0.89
Crime - Family	7.60	0.82
Biography - Sci. Fi.	9.96	0.78
Drama - Animation	1.49	0.77
Horror - Thriller	7.15	0.27
Drama - War	12.23	0.23
History - Biography	38.28	0.23
Drama - History	0.51	-0.002

Pairwise classification was accurate: the overall error rate was less than 12.4%, and for no pair of classes was higher than 38.2%.

Note that there can be overlap between two categories. For example, a movie can belong both to the genres of action and drama. In our experiment, we train and tested only on movies that belong to one genre of the pair being tested, but not the other.

6.5 Ontology Classification Fitness: Conclusions

Our method of translating tests of ontology properties into tests on an aligned textual corpus allowed us to quantitatively assess a complex property of the ontology: can it be used to cluster search results in a manner that would be recognized by experts in the domain. We translated this test on the ontology to a test on the aligned textual corpus. The results indicate that overall the ontology provides accurate classification (average F-measure of 0.49 vs. 0.16 for random clustering). Importantly, the method provides specific feedback on pairs of classes in the ontology – using pairwise classification, we can measure *a priori* high confusion between pairs of classes (*e.g.*, drama vs. history).

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

7 Evaluating the Ontology Coverage of Query Topics

7.1 Objective and Hypothesis

In this section, we suggest a methodology to deal with the situation where the ontology contains a vast number of classes which lack hierarchical structure (flat and wide structure). From the search perspective, this situation is undesirable since it is most likely unbalanced. From the evaluation perspective, applying the classification-based evaluation method discussed above directly on such flat and wide relations will not be effective, because text classification is not effective on a large number of small categories.

We illustrate this issue with the case of keywords associated to movies. Keywords in IMDb and accordingly in our Entertainment ontology are based on what Szomszor *et al.* [24] call *free-for-all* tagging. Users can add new keywords, which are then moderated to prevent spamming. Overall, there are 10,529 unique keywords in the ontology. The relation between keywords and films is *many-to-many*: there are many keywords per movie, too many or too few (as few as *one*) movies per keyword and many of the keywords may be only weakly connected to the content of the movie.

Take for example the keywords associated with the movie *"Bonnie and Clyde"*: `bank robbery, celebrity criminal` may be good search terms; but `old woman, joke, face slap` or `marriage`, intuitively, do not characterize the desired plot. Bad search terms are either too general or too specific. Over-specific terms might be useful in searching for a particular movie (*e.g.,* "I'm looking for that movie with snakes and planes and forgot its name") but, most likely, do not help exploratory search.

Hence, for search purposes, we wish to cluster keywords into more manageable and more useful exploratory dimensions. Once we cluster these keywords into a small number of topics, we can use the same classification-based evaluation method to test the quality of these clusters.

In a more abstract manner, we relate to a specific property of the ontology (a wide and shallow relation such as keywords) as the reflection of a hidden property in the domain (a smaller set of unlabeled topics). While the concrete ontology relation does not help directly the search engine for exploratory purposes, the underlying hidden property may provide quantifiable benefits for exploratory search. Again, our method leads us to translate this hypothesis into practical tests on our aligned textual corpus.

7.2 Experiment

We applied a clustering method based on Latent Dirichlet Allocation (LDA) [18] to cluster the keywords in our application. LDA is a generative probabilistic model for collections of discrete data. The main assumption of this model is that a document (represented as a bag-of-words) is a mixture of topics, and each word is generated by a topic with some probability. The mixture proportion of topics for each document as well as the topic to word generation probabilities words

are learned from a given corpus in an unsupervised process. Three parameters (K, α, β) control the number of topics, the sparsity of the document to topics distribution and the sparsity of the topic to words distributions, respectively.

LDA has been used successfully in many text mining applications. In our application, we adopt LDA to learn possible topics from a set of keywords. The topics learned by LDA are clusters of keywords. To this end, we consider the list of keywords linked to a film to form a document, and perform LDA on all such documents.

Examples of the keyword clusters we acquired are shown in Fig.1 and Fig.2. We configured the LDA process to construct 100 clusters to partition the range of over 10,000 distinct keywords found in the ontology. On average, the obtained clusters contained 20 keywords.

7.3 Results

To test the validity of the clusters obtained through the LDA analysis, we used the same method as described in Section 6. We applied the procedure to the classification induced by the keyword cluster dimension. The classifiers were trained on the reviews corpus.

As a baseline, we applied a very simple keyword clustering method – based on word similarity (two keywords are attached to the same cluster if they include a common word or words within a small edit distance. With such a simple clustering heuristic, the average F-measure obtained on text classification was extremely low (Average F-Measure = 0.07).

When using the LDA clustering method, results improved significantly:
Average F-measure is 0.48
Average MCC-measure is 0.28
Average error rate 0.26

Table 5 shows the results of the text classification method for some classes. While the classification results are not very high (as expected for the extremely noisy data set), they are much better than the baseline (range of 0.65 to 0.75 vs. 0.07). The MCC and F-measure values allow us to filter unreliable keyword clusters and to compare the potential of each keyword cluster to help in exploring the dataset.

8 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 individuals. We have demonstrated how to operate such an 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. On the basis of this dataset, we demonstrated the method on three increasingly abstract tests on the ontology: we assessed how

class	F	MCC
10	0.74	0.27
72	0.71	0.25
83	0.70	0.21
24	0.66	0.19
90	0.66	0.38
71	0.66	0.39
45	0.66	0.21
36	0.65	0.43

Table 5. One-to-all Classification of Keywords

10 flashback-sequence mother-son-relationship hotel father-son-relationship restaurant face-slap premarital-sex bar car-accident hospital funeral los-angeles-california friendship drunk-scene beach blockbuster profanity title-spoken-by-character narration cemetery

72 based-on-novel character-name-in-title independent-film number-in-title acronym-in-title lost-film hobo kilt scottish-accent entire-title-is-capitalized-acronym clock-watcher party-lifestyle team-owner essex-wife wags team-captain wives-and-girlfriends aids once-upon-a-time-in-the-title m-a-s-h

83 based-on-novel character-name-in-title independent-film circus carnival clown amusement-park criminal-justice number-in-title acrobat gypsy midget roller-coaster fortune-teller sideshow hypnotism trapeze elephant carousel ferris-wheel

Fig. 1. LDA sets with highest F-measure

the ontology captures important terminology in the domain, how it supports clustering search results along coarse relations such as genre and how it supports clustering search results along abstract relations such as topic.

We first measured the ontology *coverage of query terms*. We have found specifically that our ontology has wide coverage but lacks support for ambiguity resolution and terminological variation handling. We used techniques of fuzzy string matching and statistically trained named entity recognition to identify specific ontology terms that must be disambiguated.

Our classification experiment, measures the adequacy of the ontology to support exploratory search along coarse relations such as genre. 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. We used techniques of statistically trained text classification in these experiments.

Finally, in a topic experiment, we investigated how the noisy data provided by a cloud of unedited keywords acquired automatically into the search ontology can still provide useful classification results to help in exploratory search. We used the technique of topic modeling (specifically, the LDA algorithm) to translate a wide and shallow ontology relation (keywords) to a set of topic clusters that are then applied to cluster search results.

87 kids-and-family cartoon looney-tunes merrie-melodies australia bugs-bunny australian popeye porky-pig daffy-duck part-live-action chicken william-tell-overture woody-woodpecker sylvester pig screen-song breaking-the-fourth-wall duck anvil

86 boxing baseball sport soccer american-football football basketball coach boxer training olympics golf athlete college competition dandy stadium reverse-the-polarity-of-the-neutron-flow early-sound locker-room

36 native-american murder horse sheriff cowboy revenge gold outlaw spaghetti-western saloon ranch actor-shares-first-name-with-character bank-robbery desert cattle bandit texas shootout stagecoach arizona

Fig. 2. LDA sets with highest MCC-measure

Our tests indicate that the proposed method of translating tests on the ontology into tests on an aligned text corpus provides useful feedback to information architects that can be used to directly improve the quality of the search ontology.

Acknowledgments

This research is supported by Deutsche Telekom at the BGU T-Lab laboratories of Ben-Gurion University.

References

1. Tunkelang, D.: Faceted Search. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan and Claypool Publishers (2009)
2. Marti A. Hearst, Preston Smalley, C.C.: Faceted metadata for information architecture and search. (CHI 2006 Course)
3. White, R.W., Roth, R.A.: Exploratory Search: Beyond the Query-Response Paradigm. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan and Claypool Publishers (2009)
4. Baeza-Yates, R., Ciaramita, M., Mika, P., Zaragoza, H.: Towards semantic search. In: Natural Language and Information Systems. (2008) 4–11
5. Hearst, M.A.: Uis for faceted navigation: Recent advances and remaining open problems. In: in the Workshop on Computer Interaction and Information Retrieval, HCIR 2008. (2008)
6. Moritz Stefaner, Sebastian Ferre, S.P.J.K., Zhang, Y.: User Interface Design. In: Dynamic Taxonomies and Faceted Search : Theory, Practice, and Experience. Volume 25 of The Information Retrieval Series. Springer (2009)
7. Stoica, E., Hearst, M.A., Richardson, M.: Automating creation of hierarchical faceted metadata structures. In Sidner, C.L., Schultz, T., Stone, M., Zhai, C., eds.: HLT-NAACL, The Association for Computational Linguistics (2007) 244–251
8. Gulla, J.A., Borch, H., Ingvaldsen, J.: Ontology learning for search applications. In: Proceedings of the 6th International Conference on Ontologies, Databases and Applications of Semantics. (2007)
9. Mimno, D., McCallum, A.: Organizing the oca: learning faceted subjects from a library of digital books. In: JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, New York, NY, USA, ACM (2007) 376–385

10. 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. (2008)
11. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modelling ontology evaluation and validation. In: ESWC 2006. LNCS. Volume 4011. Springer, Heidelberg (2006) 140–154
12. 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 (2005) 51–58
13. Gomez-Perez, A.: Evaluation of ontologies. International Journal of Intelligent Systems **16** (2001) 391–409
14. Hearst, M.A.: Search User Interfaces. Cambridge University Press (2009)
15. McGuinness, D.L., van Harmelen, F.: Owl web ontology language overview. Technical Report REC-owl-features-20040210, W3C (2004)
16. Aula, A.: Query formulation in web information search. In: Proceedings of IADIS International Conference WWW/Internet. (2003) 403–410
17. Strasunskas, D., Tomassen, S.: Empirical in-sights on a value of ontology quality in ontology-driven web search. In: On the Move to Meaningful Internet Systems 2008: CoopIS, DOA, ODBASE, GADA, and IS. Springer-Verlag, Monterrey, Mexico (2008)
18. Blei, D.M., Ng, A.Y., Jordan, M.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
19. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2003) 39–48
20. Kudo, T., Matsumoto, Y.: Fast methods for kernel-based text analysis. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan, Association for Computational Linguistics (2003) 24–31
21. White, R.W., Muresan, G., Marchionini, G., eds.: ACM SIGIR Workshop on "Evaluating Exploratory Search Systems", Seattle (2006)
22. 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 (1998) 2–7
23. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Second edn. Morgan Kaufmann, San Francisco (2005)
24. Szomszor, M., Cattuto, C., Alani, H., O'Hara, K., Baldassarri, A., Loreto, V., Servedio, V.D.: Folksonomies, the semantic web, and movie recommendation. In: Proceedings of the 4th European Semantic Web Conference, Bridging the Gap between Semantic Web and Web 2.0, Innsbruck, Austria (2007)