ESAIR´08

ECIR'08 Workshop on:

Exploiting Semantic Annotations for Information Retrieval

http://www.yr-bcn.es/esair08

Glasgow, Sunday 30th of March.

ESAIR'08: Exploiting Semantic Annotations in Information Retrieval

The goal of this workshop is to create a forum for researchers interested in the use of semantic annotations for information retrieval. By semantic annotations we refer to linguistic annotations (such as named entities, semantic classes, etc.) as well as user annotations such as microformats, RDF, tags, etc. The aim of this workshop is not semantic annotation itself, but rather the applications of semantic annotation to information retrieval tasks such as ad-hoc retrieval, classification, browsing, textual mining, summarization, question answering, etc.

In the recent years there has been a lot of discussion about semantic annotation of documents. There are many forms of annotations and many techniques that identify or extract them. As NLP tagging techniques mature, more and more annotations can be automatically extracted from free text. In particular, techniques have been developed to ground named entities in terms of geo-codes, ISO time codes, Gene Ontology ids, etc. Furthermore, the number of collections which explicitly identify entities is growing fast with Web 2.0 and Semantic Web initiatives.

Despite the growing number and complexity of annotations, and despite the potential impact that these may have in information retrieval tasks, annotations have not yet made a significant impact in Information Retrieval research or applications. Further research is needed before we can unleash the potential of annotations!

Program Chairs:

Omar Alonso, A9.
Hugo Zaragoza, Yahoo! Research.

Program Committee:

John Atkinson, Universidad de Concepcion
Jamie Callan, Carnegie Mellon University
Arjen de Vries, CWI
Michael Gertz, University of California
Marko Grobelnik, Institute Jozef Stefan
Peter Jackson, Thomson Corporation
Aaron Kaplan, Xerox Research
Mounia Lalmas, Queen Mary, University of London
Hang Li, Microsoft Research
Peter Mika, Yahoo! Research
Inderjeet Mani, Brandeis University/MITRE
Mark Stevenson, University of Sheffield
Anne-Marie Vercoustre, Inria-Rocquencourt

Program & Table of Content:

Training-less Ontology-based Text Categorization

Maciej Janik and Krys Kochut

Large Scale Distributed Information Systems Lab (LSDIS)
Department of Computer Science, University of Georgia
410 Boyd Graduate Studies Research Center, Athens, GA 30602-7404
{janik, kochut}@cs.uga.edu

Abstract

We present a new, ontology-based approach to the automatic text categorization. An important and novel aspect of this approach is that our categorization method does not require a training set, which is in contrast to the traditional statistical and probabilistic methods that require a set of pre-classified documents in order to train the classifier.

In our approach, the ontology, which holds the schema, including the domain entities organized into categories and interconnected by relationships, as well as instances and linkages among them, effectively becomes the classifier for the categories of the domain concepts. After a document is converted into a thematic graph of entities, the ontological classification of the entities in the graph is then analyzed in order to determine the overall categorization of the thematic graph, and as a result, of the document.

In presented experiments, we used an RDF ontology constructed from the full English version of Wikipedia, a Web-based encyclopedia. The experiments, conducted on a collection of news articles, show that our training-less categorization method has achieved a satisfactory overall accuracy, in one experiment nearly identical to a selected traditional categorization method.

1. Introduction

Automatic text categorization is a task of assigning one or more pre-specified categories to an electronic document, based on its content. Nowadays, text classification is extensively used in many contexts. One of the examples is the automatic classification of incoming electronic news into categories, such as entertainment, politics, business, sports, etc. Standard categorization approaches utilize statistical or machine learning methods to perform the task. Such methods include Naïve Bayes [14], Support Vector Machines [27], Latent Semantic Analysis [7] and many others. A good overview of the traditional text categorization methods is presented in [22]. All of these methods require a training set of pre-classified documents that is used for classifier training; later, the classifier can correctly assign categories to other, previously unseen documents.

However, it is often the case that a suitable set of well categorized (typically by humans) training documents is not available. Even if one is available, the set may be too small, or a significant portion of the documents in the training set may not have been classified properly. This creates a serious limitation for the usefulness of the traditional text categorization methods.

As described by the World Wide Web Consortium (W3C), ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject area of knowledge, such as medicine, real estate, automobile repair, or financial management). More specifically, ontology is a data model that represents a set of concepts (entities) within a given domain and the relationships between those concepts. It is used to reason about the concepts within that domain.

In this paper, we introduce a novel text categorization method based on leveraging the existing knowledge represented in a domain ontology. The novelty of this approach is that it is not dependent on the existence of a training set, as it relies solely on the entities, their relationships, and the taxonomy of categories represented in the ontology.

In the proposed approach, the ontology effectively becomes the classifier. Consequently, classifier training with a set of pre-classified documents is not needed, as the ontology already includes all important facts.

The proposed approach requires a transformation of the document text into a graph structure, which employs entity matching and relationship identification. The categorization is based on measuring the semantic similarity between the created graph and categories defined in the ontology.

2. Motivation

An *ontology* is defined as "*an explicit specification of a conceptualization*" [10]. An ontology created for a given domain includes a set of concepts as well as relationships connecting them within the domain. Collectively, the concepts and the relationships form a foundation for *reasoning* about the domain.

Within the area of computing, the ontological concepts are frequently regarded as *classes* which are organized into hierarchies. The classes define the types of *attributes*, or properties common to individual objects within the class. Moreover, classes are interconnected by *relationships*, indicating their semantic interdependence (relationships are also regarded as attributes) [24]. Class hierarchies and class relationships form the *schema level* of the ontology, while the individuals (object instances or just instances) and links among them (relationship instances) form the so called *ground level* of the ontology. RDF/S [5] and OWL [18] are two examples of popular ontology specification languages.

A comprehensive, well populated ontology with classes and relationships closely modeling a specific domain represents a vast compendium of knowledge in the domain. It is only natural to expect that having such a comprehensive knowledge about the domain, one should be well-equipped to create software systems implementing a variety of tasks concerning the domain of the ontology. Recently, ontologies have been used in various semantic applications, ranging from business analytics [23] to semantic data integration [6].

We believe that the knowledge represented in such a comprehensive ontology can be used to identify topics (concepts) in a text document, provided the document thematically belongs to the domain represented in the ontology. Furthermore, if the concepts in the ontology are organized into hierarchies of higher-level categories, it should be possible to identify the category (or a few categories) that best classify the content of the document.

As an example, let us assume that we have a well-defined and comprehensive ontology containing knowledge about a variety of disciplines of sports in the United States, including baseball, (American) football, basketball, golf, and others. We will assume that the ontology includes a wide variety of concepts of each sport, such as a *home run*, *pitch*, *inning*, *hitter*, *quarterback*, *touchdown*, and so on, relationships, specifying that a baseball game is *composed of* innings, a punt is an *element of* the game of football and so is the *free throw* of the game of basketball. Furthermore, let us assume that the ontology contains all the relevant instances, such as sports teams (Boston Red Sox, Cleveland Indians), players, as well as the coaching and managing staffs of each of the teams, and links among them, for example specifying that one of the infielders for the Red Sox is Dustin Pedroia, and that Rafael Betancourt plays as pitcher with the Cleveland Indians. We will also assume that our ontology classes are organized into a hierarchy of higher level classes that group our concepts and instances into a number of broad categories, such as Major League Baseball, Baseball leagues, Baseball, and so on.

Now, consider a news article describing a typical baseball game. We believe that such an article most likely contains a number of occurrences of concepts and/or individuals represented in our ontology (such occurrences are known as *named entity occurrences*). There may be many clues that the document is, in fact, about baseball. First, we may be able to identify several named entity occurrences in the text of the document. For example, in the following document fragment:

> *In the seventh **inning**, **Red Sox** rookie second baseman **Dustin Pedroia** **hit** a two-run **home run** off of **Rafael Betancourt** that drove **Boston**'s **Fenway Park** wild. **Boston** scored a total of 6 **runs** in a crazy eighth **inning**, on a **single** by J.D. Drew, a three-run **double** by **Pedroia**, and a two-run **Kevin Youkilis** **home run** which bounced off a large Coke bottle advertisement. **Betancourt** gave up the first four **runs**, with the **home run** allowed by Jenson Lewis. **Youkilis** was the first **batter** he faced in relief of **Betancourt**.*

we could identify several named entities (represented here by underlined words and phrases). In addition, on the basis of our ontology, we could establish a number of relationships among the identified entities, such as the facts that Dustin Pedroia is a second baseman for Red Sox, and Kevin Youkilis is a batter for the same team. Also, we could discover that the entities (concepts) home run and inning are associated with baseball and Red Sox in an American baseball team.

Based on the above information, we could construct a *semantic graph* represented by the entities in the document and the relationships identified in the ontology. The semantic graph, which represents the thematic content of the document, could then be used to determine the document's category, or perhaps identify its topics.

Note, that the same phrase (or a single word) may identify a number of distinct entities in the ontology. For example, Betancourt (last name), recognized in the above text, may refer to the Rafael Betancourt playing for Boston Red Sox, Cuban baseball player Danny Betancourt, or perhaps Agustín de Betancourt, a structural engineer and educator from nineteenth century. It is even possible that some phrases used in the document would match many completely different

entities, not belonging to the same domain (or sub-domain). As a consequence, we could create more than one semantic graph for the entities identified in the document. Each semantic graph would offer a plausible, different *interpretation* of the thematic content of the document. If so, a semantic graph which would present the best fit with the ontology would be selected as the *dominant semantic graph*.

Finally, the dominant semantic graph could be used to establish the overall category of the document, in that we might be able to identify one, or perhaps a small number of categories that classify all, or most of the entities and relationships in the dominant semantic graph.

Another important observation is that in one of the sentences in the above text: "In the seventh inning, Red Sox rookie second baseman Dustin Pedroia hit a two-run home run off of Rafael Betancourt that drove Boston's Fenway Park wild," we could identify not only the named entities, but we might even be able to recognize the direct relationship "is second baseman of" between Red Sox and Dustin Pedroia. Such relationships recognized directly in the document, perhaps with the use of Natural Language Processing (NLP), could be used to strengthen the degree to which the semantic graph fits within our ontology.

We are approaching a time when comprehensive ontologies will be available for numerous domains. As of today, several interesting ontologies have been created in the area of biology [3], medicine [8] and culture [17]. Work is in progress on creating an ontology based on Wikipedia[1], Web encyclopedia. An RDF version of Wikipedia described in [2] is an interesting intermediate step towards this goal. The information found in such a Wikipedia-based ontology can be regarded as a source of comprehensive encyclopedic knowledge on just about any domain, ready for supporting semantics-based applications. We believe that automatic, training-less text categorization is an important example of such applications.

3. Related work

External or background knowledge can significantly improve text categorization, especially for short or ambiguous documents. It helps to unify the vocabulary, match important phrases, strengthen co-occurrences, or use related information not included in the original document in order to perform document categorization.

One of the best known sources of external knowledge is WordNet[1] – a network of related words, that can be used to match similar words and treat them as the same in classification process. One possible approach of utilizing WordNet in text classification is described in [20].

Ontologies offer knowledge that is organized in a more structural and semantic way. Their use in text categorization and topic identification has lately become an intensive research topic. As ontologies provide named entities and relationship between them, an intermediate categorization step requires matching terms to ontological entities. Afterwards, an ontology can be successfully used for term disambiguating and vocabulary unification, as presented in [4]. Another approach, presented in [16], reinforces co-occurrence of certain pairs of words or entities in the term vector that are related in the ontology. The use of descriptions of neighboring entities to enrich the information about a classified document is described in [9]. Interesting approach, although very different, is presented in [29], where authors automatically build partial ontology from the training set to improve keyword-based categorization method. Other categorization approaches based on using recognized named entities are described in [25] and [11].

Initial work has been done lately in using Wikipedia for categorization purposes. These approaches utilized the fact that Wikipedia contains a vast amount of knowledge that is interconnected and categorized. Pages in Wikipedia can be treated as named entities and categories form a kind of thesaurus that is a mixture of taxonomy and collaborative tagging [28]. Although the category graph cannot be directly transformed into a taxonomy, the work presented in [19] shows some solutions to overcome this issue. Authors also describe a method for creating additional taxonomic relations between instance entities, directly from the entity descriptions.

The analysis presented in [30] shows that Wikipedia resources can be successfully used for various NLP and categorization tasks. Semantic relatedness presented in [26] can replace WordNet in classification and even outperform it. Finally, Wikipedia's category network can be used to identify document topics, as described in [21]. This approach utilizes statistical methods based on the similarity of phrases in the document to entity names, and later, their category assignment.

4. Training-less text categorization

The proposed categorization method relies on converting the analyzed text into a semantic graph based on the ontological knowledge, and later finding categories that closely describe the constructed graph in terms of coverage of the entities in the graph, especially focusing on the core entities in the graph as

[1] *http://en.wikipedia.org*

well as the height of the covering categories in the category hierarchy.

We assume that the domain ontology used for the purpose of text categorization has a rich instance base of interconnected entities (with proper labels) that can be used for spotting them in the analyzed text. The entities are classified according to a taxonomy that will be used for categorization purposes. The target classification categories are defined as a taxonomy sub-hierarchy, list of related classes or mix of both the above. We also assume that the analyzed text is related to the knowledge domain represented in the ontology.

The outline of the categorization algorithm is presented below. The algorithm has two distinct phases: the construction of the semantic graph and its classification. The details of each step of the algorithm are explained in detail later in this section.

Semantic graph construction

1. Identify all named entities in the text of the document using different name labels in ontology associated with them and assign initial weights to the entities, based on the strength of each match; the entities are the nodes of the initial semantic graph.
2. Add the edges connecting the spotted entities, based on the relationships present in the ontology and establish the connectivity weights based on the importance of the relationship in the ontology schema.
3. Propagate and recalculate the weights of entities in the created graph; locate the entities with the highest weights, which are called authoritative entities.

Thematic graph identification and classification

4. Identify the dominant thematic graph, the largest and most important connected component of the semantic graph for further analysis.
5. Identify the central and authoritative entities in the dominant thematic graph.
6. Assign ontology categories to entities in the dominant thematic graph, based on the taxonomy categories included in the ontology schema.
7. Identify the target classification categories that (i) include the authoritative and the central entities, (ii) cover the largest part of the component, (iii) are closest to the graph entities in terms of their height in the category hierarchy.
8. The identified categories represent the classification of the document.

We now present our algorithm in detail. Example document, created relevant thematic graph and categorization result is presented in the appendix.

4.1. Semantic graph construction

The first step in preparing the text for ontology-based classification is the construction of the semantic graph, based on the text of the document. The purpose of having a semantic graph is to shift the analysis focus from the words, strings, and phrases occurring in the document to the entities and semantic relationships among them.

We will assume that word stemming and stop words removal may be applied to the document text before entity identification step. The ontology entities occurring in the analyzed document are identified by matching document phrases with entity literals (used as entity names) stored in the ontology. Such literals are usually represented as the values of certain properties associated with the entity and used as its identification. We assume that these properties define the entity name (usually known as its label), and may also specify the entity name's synonyms (aliases). We assign the *weight of an entity match* based on which of the identification properties was used in the match. We give preference to an exact match to the entity's label.

An entity name can be matched in several places in the document. It is important information, which is analogous to the term frequency used in the traditional text categorization methods. Such a multi-occurrence entity match is reflected by an increased weight of the entity. However, in order to limit a drastic increase in the weight of a frequently occurring entity, we use the following formula to establishing the initial weight of each entity:

$$w = 1 - \frac{1}{1 + \sum_{i=1..n} p_i * s_i}$$

In the formula, w is the initial entity weight and n represents the number of matches for the entity. The term p_i represents the weight of the identification property connecting the matched literal (name or alias) to the entity in the i-th match, and s_i is the measure of the similarity between the matched literal and text phrase, taking into account any differences introduced by word stemming and/or stop word removal. In case the entity identification process does not involve stemming and stop words removal, s_i set to 1.

Note, that a matched literal may point to multiple entities in the ontology, since different entities may have the same names or aliases. Therefore, the number of identified entities may be higher than the number of matched phrases in text. Many of them may be incorrectly identified (false positives), and will be eliminated later. However, at this stage, all of the

identified entities are used as nodes in the semantic graph being constructed.

Since all of the identified entities are represented as concepts or individuals in the ontology, the ontology may contain relationships connecting many pairs of them. Such existing relationships are added as edges to the identified entities (nodes) in order to form the *semantic graph* for the document.

The addition of the relationships into the semantic graph is a very important step in determining the categorization of the document. In fact, we view this step as the addition of the domain knowledge, represented in the ontology, in order to connect the discrete concepts (entities) in the document to form semantically related graph regions. The added ontological knowledge, even though it may not have been directly represented in the document, offers plausible semantic interpretations for co-occurrence of these entities. These semantic interpretations form the key information in determining the document classification.

Our categorization algorithm concentrates on most important and most central entities in the analyzed document. To recognize most important entities we utilize the hubs and authorities algorithm [13]. It helps to reinforce entities that are important according to ontology, even if they were underrepresented in the original text. In this approach we can also weigh different named relationships differently, in order to increase or decrease their importance reflecting the importance of their semantics. Such weights can be assigned according to the relationship rarity or other schema chosen by user.

4.2. Thematic graph and core entities

It is possible that the analyzed document covers more than one topic. In addition, during the entity matching phase, many entities may have been added to the semantic graph even though they are unrelated or, perhaps, weakly related to the main topic of the document. Furthermore, some phrases in the document might have led to the identification of multiple entities, but, perhaps, only one of them represents the proper match within the context of the document.

This step of the algorithm involves the selection of a sub-graph of the previously constructed semantic graph which represents the best interpretation of the recognized entities and relationships. We call such a sub-graph the *thematic graph*. The selection of the thematic graph is based on the assumption that the entities within one topic are related to each other, forming a connected component in the semantic graph. The semantic graph is created using the entities and relationships from the ontology, therefore the entities and relationships in that component should fall within one topic (category). Entities in the semantic graph that are not connected to other entities, or that belong to other, perhaps smaller connected components most probably belong to other topics.

If a given document is focused on specific topics (which is the assumption of automatic text categorization), there should be a single or just very few *dominant thematic graphs* in the document's semantic graph that correspond to main topics of the document. For further analysis and categorization, we select a thematic graph that has the largest number of instances and has the largest total of entity weights. In case a few thematic graphs have very similar scores, all of them are included for further analysis. If more than one thematic graph has been selected, it can mean that the document is focused on more than one topic.

The selection of the dominant thematic graphs effectively eliminates the entities unrelated to the main topics of the document, such as incorrectly selected entities, or ambiguous entities that share the same name. Furthermore, the graph reduction entails the removal of *satellite* (or *fringe*) entities that are weakly related to dominant thematic graph. This step reduces the number of low-value information, decreases the level of noisy information, and enables to shift the analysis to the core topics of the document.

Furthermore, we compute the centrality score of the entities in the thematic graph in order to find the most central entities as *topic landmarks*. In our experiments, we used geodesic closeness measure to find most central entities. The geodesic closeness measure is defined as the reciprocal of the sum of the shortest paths between the selected vertex and all other vertices in the component:

$$Centrality(v_i) = \frac{1}{\sum_j d(v_i, v_j)}$$

where $d(v_i, v_j)$ is the shortest path distance in between vertices v_i and v_j (here, we treat the thematic graph as an undirected graph).

The calculation of the authorities and the centrality measure results in locating the core entities in the graph. The best authorities and the most central entities are selected as the core of the thematic graph. They are determined to be the most relevant to the document topic. Note, that the best authorities do not have to be the most central entities, and vice versa.

Selection of core entities should include both the best authorities and the most central entities. This ensures that the topic landmarks and important entities will be

included in the categorization step. The selection can include a certain percent of all entities from the thematic graph, or finding a good cut-off point. In our experiments, we decided to include up to 10% of all entities in the thematic graph the core entities from both groups. We also set minimum of 3 entities from each group to assure presence of most central and most important entities in graph core.

4.3. Thematic graph categorization

The categorization process shifts the attention from the dominant thematic graph, an instance level graph composed of the matched entities and relationships, to the taxonomy represented in the ontology schema. Each entity in the selected dominant thematic graph has its importance weight and almost each one of them has assigned at least one class in the taxonomy.

Finding a category that offers the best fit for whole thematic graph (or its part) is an optimization of a number of different, possibly conflicting objectives. The best category should:

- cover (be a class or super-class of) the highest number of entities in the thematic graph,
- be the lowest level category (in terms of the hierarchy of categories), and
- include the highest number of the core entities.

The category coverage of the thematic graph is illustrated in Figure 1.

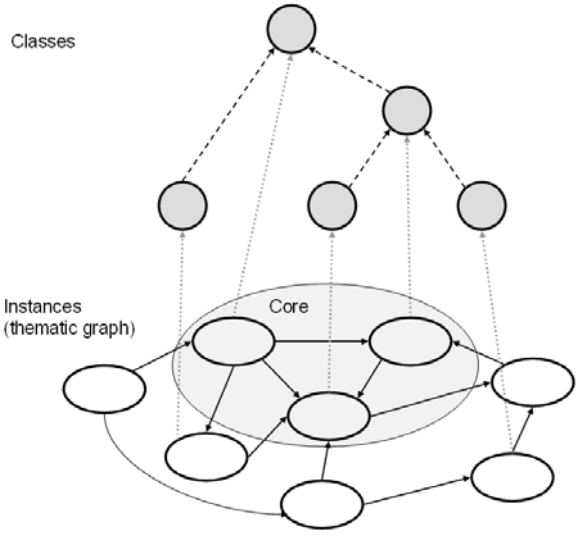

Figure 1. Thematic graph and categories

The selected class should offer the best fit for whole thematic graph (or its large part). Taking into consideration the properties of the best coverage class, we use the following formula for calculating the class score:

$$s_{Ci}(h_{\max}) = 1 - (1 - \frac{1}{1 + \sum_j \frac{w_j}{h(C_i, e_j)^2} + \sum_k \frac{w_k}{h(C_i, e_{Ck})}})$$

where s_{Ci} is the categorization score for class C_i, that includes reachable entities e up to depth h_{max} ; e_j and e_{Ck} represent respectively entity and core entity reachable up to depth h_{max} from class C_i ; w_j and w_{Ck} are weights of entity e_j and core entity e_{Ck} ; $h(C_i, e)$ is the hierarchical distance between category C_i and the covered entity e. The first summation is over all of the entities reachable from category C_i up to depth h_{max}, while second include only core entities.

Any class which does not cover at least one of the core entities is rejected, as it is not associated with the main topic of the document. The remaining classes, ranked according to their scores, represent the categorization of the document, relatively to the taxonomy in the ontology schema. At this step document has assigned multiple ranked categories that describe its content.

The final step of the document categorization requires matching between the internal categorization and user-defined topic categories. There are many possible approaches to match the assigned taxonomy classes with selected topics. We suggest following different approaches that depend on the method used in defining the topic categories.

The proposed score computation favors the classes that are closest to the thematic graph, not taking into account their depth in the taxonomy. In case the final topics are defined as parts of the taxonomy, we should favor matching to the lowest category in the hierarchy as the most specific one. Final categorization score s_{Ci} for class C_i should be increased as the depth of the matched class increases. To capture user's interest in matching hierarchy, score s_{Ci} can increase linearly, exponentially or in other preferred method with the depth of class C_i.

In case the class hierarchy is similar to that found in Wikipedia, where the defined categories instead of a clear taxonomy form a thesaurus, such an approach will likely give unsatisfactory results. In a thesaurus, the further we move away from the original category, the less relevant the matched category becomes. The categorization score should be modified to incorporate the relational (thesaurus-like) distance between classes.

Assigning external category is based on matching modified highest ranked classes to category definition. Starting from the class with the highest score, assign it to all appropriate categorization topics and increase their weight accordingly. Process can continue for all found categories, or only include selection of top k classes, or until one topic has dominant score.

5. Experiment design

In our experiments, we used the RDF ontology created from the English version of Wikipedia, using a slightly modified DBpedia approach [2] and text corpora of news articles gathered from the CNN Web site (www.cnn.com). Our implementation used Brahms as the backend RDF storage for the ontology [12]. We compared the accuracy of our training-less categorization method with one of the traditional, text categorization methods implemented by the BOW toolkit [15]. The BOW toolkit, similarly to other traditional methods, relies on the existence of a document training set, required for training of the classifier.

5.1. Wikipedia ontology

The RDF/S ontology was derived from Wikipedia dump from 2007-09-08. It contains 2,062,198 instance entities that contribute to a highly connected graph of 67,279,865 statements, and 4,409,200 literals that can be used for entity matching. On average, each entity has assigned 2.85 literals using different relationships. The schema part has 311,908 classes (Wikipedia categories) organized in 532,191 statements, mostly describing semi-hierarchical dependencies among the classes. Each entity, on average, has been assigned to 2.64 classes.

We utilized a modified DBpedia approach to create an RDF/S ontology from Wikipedia. Our modifications related to handling of the extraction of the templates included in a typical Wikipedia entry and assignment of literal values.

In DBpedia, the included templates (except from Infoboxes) become separate entities, connected to the source page. We shortcut these links and entities mentioned in the template content are set as directly related to the source entity. As these additional entities come from named templates, they are not linked by *the href* relationship, but by a named relationship derived from the template name. These named relationships are more important in our categorization method, as they carry more specific information about the existing connections between the entities, than simple *href* links.

The literal values, such as entity names, redirections, and disambiguation are very important for creating phrases to spot entities in the document. We have created separate named relationships to distinguish among the direct names of entities (page names in Wikipedia), redirections (redirection pages), and entities names included in the disambiguation pages. Wikipedia also utilizes a convention of disambiguating entity names by adding contextual information enclosed within parentheses and listed after the entity name, e.g., “Jaguar”, “Jaguar (car)” and “Jaguar (band)”. Such full phrases do not exist in documents, as the context of the document provides enough information for human to properly disambiguate the entity. For entities with such names, we create a shorter literal by omitting the context information and add it as an alternate, shorter name, using specific property to distinguish it from the full name.

5.2. CNN text corpora

We tested the proposed categorization method on the recent CNN news articles (www.cnn.com) obtained from CNN RSS feeds between 2007-07-03 and 2007-09-04. The choice of recent news articles is related to choosing Wikipedia as our categorization ontology. Wikipedia is an encyclopedia of general knowledge that contains very recent entries. CNN articles describe facts from general knowledge and broadly defined CNN categories can relatively easy mapped to Wikipedia categories.

Our CNN text corpora is composed of 2,590 news articles assigned to 12 different categories. Each category was associated with a single RSS feed. For comparison with a traditional, probabilistic categorization method, we divided it into a 50/50 split, where the training and testing sets had 1,295 documents each. The selected categories with the split details are presented in Table 1.

Table 1 CNN text corpora details

	CNN Category	Train set	Test set
1	Education	4	7
2	Health	91	87
3	Money – autos	37	26
4	Money – companies	271	275
5	Money – taxes	15	12
6	Politics	171	167
7	Science and space	35	27
8	Sport – MLB	143	171
9	Sport – NBA	139	122
10	Sport – NFL	203	222
11	Sport – NHL	93	100
12	Travel	93	79

5.3. Category mapping: CNN and Wikipedia

The direct categorization to the ontology classes, as proposed in the description of our method, does not require supplying definition of the categories. For the purpose of evaluation of document categorization and comparison to one of the traditional methods, a

mapping between the selected CNN categories and suitable Wikipedia categories had to be created.

We decided to prepare the mapping between CNN and Wikipedia categories using a simple approach. For each CNN category, we have manually selected the main concepts from among the Wikipedia categories (roots) and added their subcategories up to the depth of 3. Depth limit was set due to fact, that the Wikipedia categories form not a taxonomy, but a thesaurus. Subcategories do not follow the strict semantics of the *rdf:subclass* property, but only are (closely) related to each other. The mapping of the roots of the selected Wikipedia categories to CNN categories and the numbers of the included subcategories in presented in Table 2.

Table 2 Wikipedia root categories for CNN classes.

CNN category	Wikipedia root classes	Number of subcategories
education	Category:Education	1467
health	Category:Health	1505
money_autos	Category:Automobiles	1350
money_companies	Category:Business Category:Economics Category:Stock_market	2509
money_taxes	Category:Accountancy Category:Taxation	146
politics	Category:Politics Category:Politicians	7746
science_and_space	Category:Science Category:Space	833
sport_mlb	Category:Baseball Category:Major_League_Baseball	1710
sport_nba	Category:Basketball Category:National_Basketball_Association	2438
sport_nfl	Category:Football Category:National_Football_League	8251
sport_nhl	Category:Hockey Category:National_Hockey_League	1858
travel	Category:Travel	714

5.4. Reference categorization method

We selected the Naïve Bayes classification method available in the BOW toolkit as a baseline for comparing categorization accuracy. We performed document categorization using two types of training sets. In the first experiment, we used as the training set a random subset of the Wikipedia entries (full pages) assigned to the categories identified in the created CNN category mapping. In this experiment BOW source of training documents differs from the categorized ones. In the second experiment, a BOW classifier was trained on the articles from CNN text corpora. This followed a traditional approach for classifier training, where the training documents come from the same source as the documents to be classified.

The selected Wikipedia categories for our CNN category mapping cover over 400,000 entries. For each CNN category, we randomly chose up to 2,000 representative pages from Wikipedia and trained BOW on them. To check the consistency of the categorization results, 10 different training sets were created and tested. We believe that this experiment offered a better comparison with the direct ontology-based classification, as both the traditional (probabilistic) and ontology-based classifiers used the same source of information for the categorization task.

6. Experiment results

We performed three types of experiments:

- Our proposed training-less ontology-based categorization with the use of Wikipedia and CNN category mapping,
- BOW categorization trained on a subset of Wikipedia articles relevant to the CNN-mapped categories, and
- BOW categorization trained on our test split of CNN articles.

Categorization of CNN articles was performed using 1,295 documents from the testing set.

Our ontology-based method that used prepared CNN category mappings reached an accuracy of 80%. No training was necessary in this case. Different runs of categorization of the CNN corpora by Naïve Bayes categorization using prepared subsets of Wikipedia documents as the training set achieved only 73% accuracy. In this test both categorization methods used the same knowledge (ontology-based method) and documents (BOW) to perform categorization. Difference in original CNN categories and Wikipedia categories required to use prepared mapping.

When BOW was trained on the training set of the CNN articles, it was able to achieve accuracy slightly over 94%. In this case training and testing document came from the same source and no intermediate mapping was used.

The detailed categorization results from all three tests are respectively presented in Tables 3, 4 and 5.

Table 3 Ontology-based categorization of CNN document split using Wikipedia ontology with prepared category mapping.

	CATEGORY	0	1	2	3	4	5	6	7	8	9	10	11	12	total	correct
0	Education	6	.	.	.	.	1	.	.	.	.	.	.	.	7	85.71%
1	Health	2	70	.	3	.	4	4	.	.	.	.	4	.	87	80.46%
2	money_autos	.	.	17	8	.	.	1	.	.	.	.	.	.	26	65.38%
3	money_companies	.	20	10	213	19	2	5	1	.	.	.	5	.	275	77.45%
4	money_taxes	.	.	.	3	8	1	.	.	.	.	.	.	.	12	66.67%
5	Politics	.	6	.	8	.	148	3	.	.	.	.	2	.	167	88.62%
6	science_and_space	1	1	.	1	.	1	21	.	.	.	.	2	.	27	77.78%
7	sport_mlb	.	2	.	6	.	3	1	153	.	.	.	6	.	171	89.47%
8	sport_nba	.	3	.	7	.	12	3	.	91	1	.	4	1	122	74.59%
9	sport_nfl	.	3	1	7	.	16	4	.	.	185	.	6	.	222	83.33%
10	sport_nh	.	.	1	7	.	3	2	.	1	3	77	6	.	100	77.00%
11	Trave	.	5	.	7	.	9	10	.	.	.	.	48	.	79	60.76%
12	Unknown	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0.00%
	Classified documents	: 1295														
	Correctly classified	: 1037														
	Achieved accuracy	: 80.077														

Table 4 Naïve Bayes categorization (BOW implementation) results of CNN document split with Wikipedia documents training set.

Correct: 1949 out of 1295 (73.28 percent accuracy); Confusion details, row is actual, column is predicted

	Classname	0	1	2	3	4	5	6	7	8	9	10	11	total	Correct
0	Education	7	.	.	.	.	.	.	.	.	.	.	.	7	100.00%
1	Health	23	55	.	1	1	.	3	.	.	.	.	4	87	63.22%
2	money_autos	.	.	23	3	.	.	.	.	.	.	.	.	26	88.46%
3	money_companies	1	11	16	169	74	.	.	.	.	.	.	4	275	61.45%
4	money_taxes	.	.	.	.	12	.	.	.	.	.	.	.	12	100.00%
5	Politics	11	4	.	2	40	100	.	.	.	.	.	10	167	59.88%
6	science_and_space	1	.	.	.	.	.	22	.	.	.	.	4	27	81.48%
7	sport_mlb	1	1	.	3	5	.	.	155	.	.	.	6	171	90.64%
8	sport_nba	2	.	.	3	4	.	.	.	99	1	.	13	122	81.15%
9	sport_nfl	13	1	.	9	7	2	.	.	8	160	.	22	222	72.07%
10	sport_nhl	5	.	1	7	5	.	.	.	2	.	75	5	100	75.00%
11	Travel	1	1	.	2	3	.	.	.	.	.	.	72	79	91.14%
	Percent_Accuracy average 73.28 stderr 0.00														

Table 5 Naïve Bayes categorization (BOW implementation) results of CNN document split with CNN training set.

Correct: 1220 out of 1295 (94.21 percent accuracy); Confusion details, row is actual, column is predicted

	Classname	0	1	2	3	4	5	6	7	8	9	10	11	total	correct
0	Education	2	.	.	.	.	2	.	.	.	.	.	.	4	50.00%
1	Health	.	80	.	4	.	2	.	.	.	2	.	3	91	87.91%
2	money_autos	.	.	16	20	.	.	.	.	.	.	.	1	37	43.24%
3	money_companies	.	1	4	263	.	2	.	.	.	.	.	1	271	97.05%
4	money_taxes	.	.	.	9	4	2	.	.	.	.	.	.	15	26.67%
5	Politics	.	2	.	.	.	169	.	.	.	.	.	.	171	98.83%
6	science_and_space	.	.	.	.	.	.	33	.	1	.	.	1	35	94.29%
7	sport_mlb	.	.	.	1	.	.	.	140	.	2	.	.	143	97.90%
8	sport_nba	.	.	.	.	.	.	.	1	135	3	.	.	139	97.12%
9	sport_nfl	.	.	.	.	.	1	.	.	.	202	.	.	203	99.51%
10	sport_nhl	.	.	.	1	.	.	.	.	2	.	90	.	93	96.77%
11	Travel	.	1	.	2	.	4	.	.	.	.	.	86	93	92.47%
	Percent_Accuracy average 94.21 stderr 0.00														

6.1. Analysis of the results

Our training-less ontology-based method achieved good results, compared to statistical method trained on Wikipedia knowledge, although when BOW was trained on source CNN articles, its accuracy was considerably higher.

We investigated the potential sources of misclassification problems in the CNN corpora. Analysis of several articles and created thematic graphs, together with the ranked categories revealed following causes of misclassifications:

- the created mapping between the CNN and Wikipedia categories were too broad and imprecise,
- the difference between the article's actual thematic content and the assigned category by CNN,
- an unevenly developed structure of Wikipedia link and category for different domains.

The imprecise mapping of CNN categories is both a result of ambiguity in defining CNN categories and the used category hierarchy in Wikipedia. In some cases, Wikipedia categories obtained by descending the Wikipedia category hierarchy were poorly related to the source category. In other cases, due to a thesaurus-like structure of Wikipedia categories, some categories were included in incorrect CNN mappings.

The second type of misclassifications is tightly related to the difference of article's actual content and a reader's *perceived* interest. It has been responsible for a larger portion of the misclassified documents. The ontology-based categorization analyzes the document content in order to create a thematic graph, and then finds its best fit into the ontological knowledge. On the other hand, the categories assigned by CNN mainly reflect the reader's perceived interest. The created Wikipedia-based ontology contains encyclopedic knowledge, which describes basic facts and their relationships. It does not favor any specific types of entities such as people, companies, or places. The (human assessed) perceived interest in the article may decide the article's category solely on a single type of high-interest entity, and not on the thematic content of the document.

As an example, consider an article about cardio-vascular health problems of a certain politician. From a reader's perspective, the article belongs to *politics*, as the politician is the main point of interest. On the other hand, the majority of the document content is about the disease, treatment, or perhaps recovery. In the analysis of the created semantic graph, the politician most probably will not become one of the core entities, and the graph core will concentrate on medical issues. This will result in the final categorization into the *health* domain.

Finally, some misclassifications were related to the ontology and Wikipedia itself. Some parts of Wikipedia are much better covered and interconnected than others. Consequently, entities from the better covered regions have a higher chance to be recognized and, due to their high connectivity, create a better thematic graph.

Focusing only on the document content, represented by entities and relationships, can be perceived both as the strength and the weakness of our categorization method. The strength comes from utilizing the background knowledge from the ontology that may not be present in the document. The weakness lies in the very difference between facts and perceived interests, which may require a much more sophisticated mapping or a modification of our algorithm to favor certain types of entities, relationships, or structures. We believe that it may be overcome by using certain NLP methods in building the thematic graph, and providing a more specific and defined context of interest in the classification step of the algorithm.

7. Conclusions and future work

In this paper we presented a novel text categorization method based on ontological knowledge that does not require a training set. The tests performed using an RDF ontology derived from Wikipedia demonstrated its effectiveness and practical value. In comparison with one of the statistical methods trained on the documents from the categorization ontology, our classification algorithm achieved nearly identical overall accuracy.

The presented approach and our experiments confirm that a rich and comprehensive ontology can be successfully used as a text classifier. The selection of a proper mapping between the ontology classes and user defined categories remains as an open question. In the near future, we plan to concentrate on defining a *categorization context* that could be used to specify perceived areas of interest for the user.

Another direction of future work is in including more semantics from the analyzed text. We plan to investigate the usefulness of NLP methods in discovering named relationships between the identified entities in the document itself. The relationships would be used for categorization in order to either strengthen the existing relationships in the knowledge base or to add additional information, not yet existing in the ontology.

References

[1] WordNet: An Electronic Lexical Database. The MIT Press (1998)

[2] Auer, S., Lehmann, J.: What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content.: European Semantic Web Conference (ESWC'07). Springer, Innsbruck, Austria (2007) 503-517

[3] Bada, M., Stevens, R., Goble, C., Gil, Y., Ashburner, M., Blake, J.A., Cherry, J.M., Harris, M., Lewis, S.: A Short Study on the Success of the Gene Ontology. Journal of Web Semantics **1** (2004)

[4] Bloehdorn, S., Hotho, A.: Text Classification by Boosting Weak Learners based on Terms and Concepts. 4th IEEE International Conference on Data Mining (ICDM'04) (2004)

[5] Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. In: McBride, B. (ed.): http://www.w3.org/TR/rdf-schema/ (10 Feb 2004)

[6] Buccella, A., Cechich, A., Brisaboa, N.R.: Ontology-Based Data Integration. In: Rivero, L.C., Doorn, J.H., Ferraggine, V.E. (eds.): Encyclopedia of Database Technologies and Applications. Information Science Reference (2005)

[7] Deerwester, S., Dumais, S., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the Society for Information Science (1990) **41** (1990) 391-407

[8] Eccher, C., Purin, B., Pisanelli, D.M., Battaglia, M., Apolloni, I., Forti, S.: Ontologies supporting continuity of care: The case of heart failure. Computers in Biology and Medicine (2006) **Jul-Aug; 36** (2006) 789-801

[9] Gabrilovich, E., Markovitch, S.: Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. 21th National Conference on Artificial Intelligence, Boston, MA, USA (2006)

[10] Gruber, T.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition **5** (1993) 199-220, 1993

[11] Hammond, B., Sheth, A.P., Kochut, K.J.: Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content. Real World Semantic Web Applications, IOS Press, 2002 (2002)

[12] Janik, M., Kochut, K.J.: BRAHMS: A WorkBench RDF Store And High Performance Memory System for Semantic Association Discovery. Fourth International Semantic Web Conference (ISWC 2005), Galway, Ireland (2005)

[13] Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. ACM-SIAM Symposium on Discrete Algorithms (1998)

[14] Lewis, D.D.: Naive (Bayes) at forty: The independence assumption in information retrieval.: ECML-98, 10th European Conference on Machine Learning, Chemnitz, DE (1998)

[15] McCallum, A.K.: Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering.: http://www.cs.cmu.edu/~mccallum/bow (1996)

[16] Nagarajan, M., Sheth, A.P., Aguilera, M., Keeton, K., Merchant, A., Uysal, M.: Altering Document Term Vectors for Classification - Ontologies as Expectations of Co-occurrence. LSDIS Technical Report (November, 2006)

[17] Ossenbruggen, J.v., Amin, A., Hardman, L., Hildebrand, M., Assem, M.v., Omelayenko, B., Schreiber, G., Tordai, A., Boer, V.d., Wielinga, B., Wielemaker, J., Niet, M.d., Taekema, J., Orsouw, M.-F.v., Teesing, a.A.: Searching and Annotating Virtual Heritage Collections with Semantic-Web Techniques. Museums and the Web 2007, San Francisco, California (2007)

[18] Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL Web Ontology Language Semantics and Abstract Syntax. http://www.w3.org/TR/owl-semantics/ (10 Feb 2004)

[19] Ponzetto, S.P., Strube, M.: Deriving a Large Scale Taxonomy from Wikipedia. Twenty-Second Conference on Artificial Intelligence (AAAI'07), Vancouver, Canada (2007)

[20] Rosso, P., Ferretti, E., Jiménez, D., Vidal, V.: Text Categorization and Information Retrieval Using WordNet Senses. 2nd Global WordNet Int. Conf., GWN-2004, Brno, Czech Republic (2004)

[21] Schonhofen, P.: Identifying document topics using the Wikipedia category network.: ACM International Conference on Web Intelligence (WI 2006), Hong Kong (2006)

[22] Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR) **34** (2002) 1 - 47

[23] Semagix: Anti-Money Laundering - CIRAS. http://www.semagix.com/solutions_ciras.html.

[24] Sheth, A.P., Arpinar, I.B., Kashyap, V.: Relationships at the Heart of Semantic Web: Modeling,

Discovering, and Exploiting Complex Semantic Relationships. In: Nikravesh, M., Azvin, B., Yager, R., Zadeh, L. (eds.): Enhancing the Power of the Internet: Studies in Fuzziness and Soft Computing. Springer Verlag (2003)

[25] Sheth, A.P., Bertram, C., Avant, D., Hammond, B., Kochut, K.J., Warke, Y.: Semantic Content Management for Enterprises and the Web. IEEE Internet Computing **July/August 2002** (2002)

[26] Strube, M., Ponzetto, S.P.: Wikirelate! Computing Semantic Relatedness Using Wikipedia.: Twenty-First Conference on Artificial Intelligence (AAAI'06), Boston, Massachusetts (2006)

[27] Vapnik, V.: The nature of statistical learning theory. Springer Verlag (1995)

[28] Voss, J.: Collaborative thesaurus tagging the Wikipedia way. ArXiv Computer Science e-prints **cs/0604036** (2006)

[29] Wu, S.-H., Tsai, T.-H., Hsu, W.-L.: Text categorization using automatically acquired domain ontology. 6th international workshop on Information retrieval with Asian languages - Volume 11, Sapporo, Japan (2003)

[30] Zesch, T., Gurevych, I.: Analysis of the Wikipedia Category Graph for NLP Application. Workshop - TextGraphs-2: Graph-based Methods for Natural Language Processing at NAACL Human Language Technologies Conference, Rochester, New York (2007)

Appendix – categorization example

Example text (downloaded from WikiNews[2]):

*The **Boston Red Sox** are once again headed to the **World Series** after being down three games to one. **The Red Sox** were most recently in the 2004 **World Series**, which they won.*

*Last night's game ended with **the Red Sox** winning 11-2 over the **Cleveland Indians**. The $103 million rookie import from Japan, **Daisuke Matsuzaka** (nicknamed "**Dice-K**"), pitched five **innings** for **Boston**, allowing two **runs** on six **hits**. Cleveland's Jake Westbrook started and took the loss. This proved to be a much better showing for **Boston**'s "**Dice-K**" than his previous outing, which **Boston** lost.*

***The Sox** jumped out to a quick lead, scoring a **run** in each of the first three **innings** on a **single** by **Manny Ramirez**, a sacrifice ground-out by **Julio Lugo**, and a **sacrifice fly** by **Mike Lowell**. The Indians scored their first **run** on a **Ryan Garko** **double** in the fourth **inning**, and a **Grady Sizemore** **sacrifice fly** in the fifth made the score 3-2 in favor of **the Sox**.*

*In the sixth **inning**, **Hideki Okajima** came in to relieve "**Dice-K**" and pitched two scoreless **innings** before **Jonathan Papelbon** came in to close in the eighth. He entered the game with runners on first and second and no outs, but quickly retired the side and in the ninth managed to maintain the nine **run** lead, once again giving fans a performance of his Riverdance style **victory dance**.*

*In the seventh **inning**, **Red Sox** rookie second baseman **Dustin Pedroia** **hit** a two-run **home run** off of **Rafael Betancourt** that drove **Boston**'s **Fenway Park** wild. **Boston** scored a total of 6 **runs** in a crazy eighth **inning**, on a **single** by J.D. Drew, a three-run **double** by **Pedroia**, and a two-run **Kevin Youkilis** **home run** which bounced off a large Coke bottle advertisement. **Betancourt** gave up the first four **runs**, with the **home run** allowed by **Jenson Lewis**. **Youkilis** was the first **batter** he faced in relief of Betancourt.*

***The Sox** will go on to face the **Colorado Rockies**, the surging National League Champions. The series will begin October 24th, with the first game at **Fenway Park**.*

In this text, underlined words and phrases where recognized as entities in Wikipedia, but only the ones in bold were selected to thematic graph for further categorization.

Created thematic graph is presented in Figure 2 on the next page. Most important and central entities are shaded in gray. Majority of relationships between selected entities are simple page references (href) represented as black arrows. Blue, bold arrows represent relationships via templates included in pages. They carry some more information than simple page references. Finally, red, bold arrows represent relationships discovered in Wikipedia's infoboxes. They are the most important connections between entities, as have specific semantic meaning, defined by infobox specification.

Most important and most central entities discovered in the thematic graph, which became the core entities for the categorization process:

- Boston_Red_Sox
- Home_run
- Run_(baseball)
- Single

After analysis of the thematic graph with special attention to the core entities, our algorithm assigned following Wikipedia categories (presented top 15):

- Category:Major_League_Baseball_teams
- Category:Sports_clubs_established_in_1901
- Category:Boston_Red_Sox
- Category:Major_League_Baseball
- Category:Sports_in_Boston
- Category:Singles
- Category:Baseball
- Category:Boston,_Massachusetts
- Category:Baseball_in_the_United_States
- Category:Sports_in_the_United_States
- Category:Baseball_teams
- Category:Sports_leagues_in_the_United_States
- Category:Sports_leagues_in_Canada
- Category:Sports_in_the_United_States_by_city
- Category:Baseball_leagues

The categories shown above are assigned using only document text and ontological knowledge. This is the categorization result of the proposed algorithm. Partial graph of Wikipedia categories associated with selected entities from the thematic graph is presented in Figure 3.

Using prepared mapping external to ontology-based categorization algorithm, the document was assigned to the CNN category *Sport MLB*.

[2] http://en.wikinews.org/wiki/Boston Red Sox win American League Championship

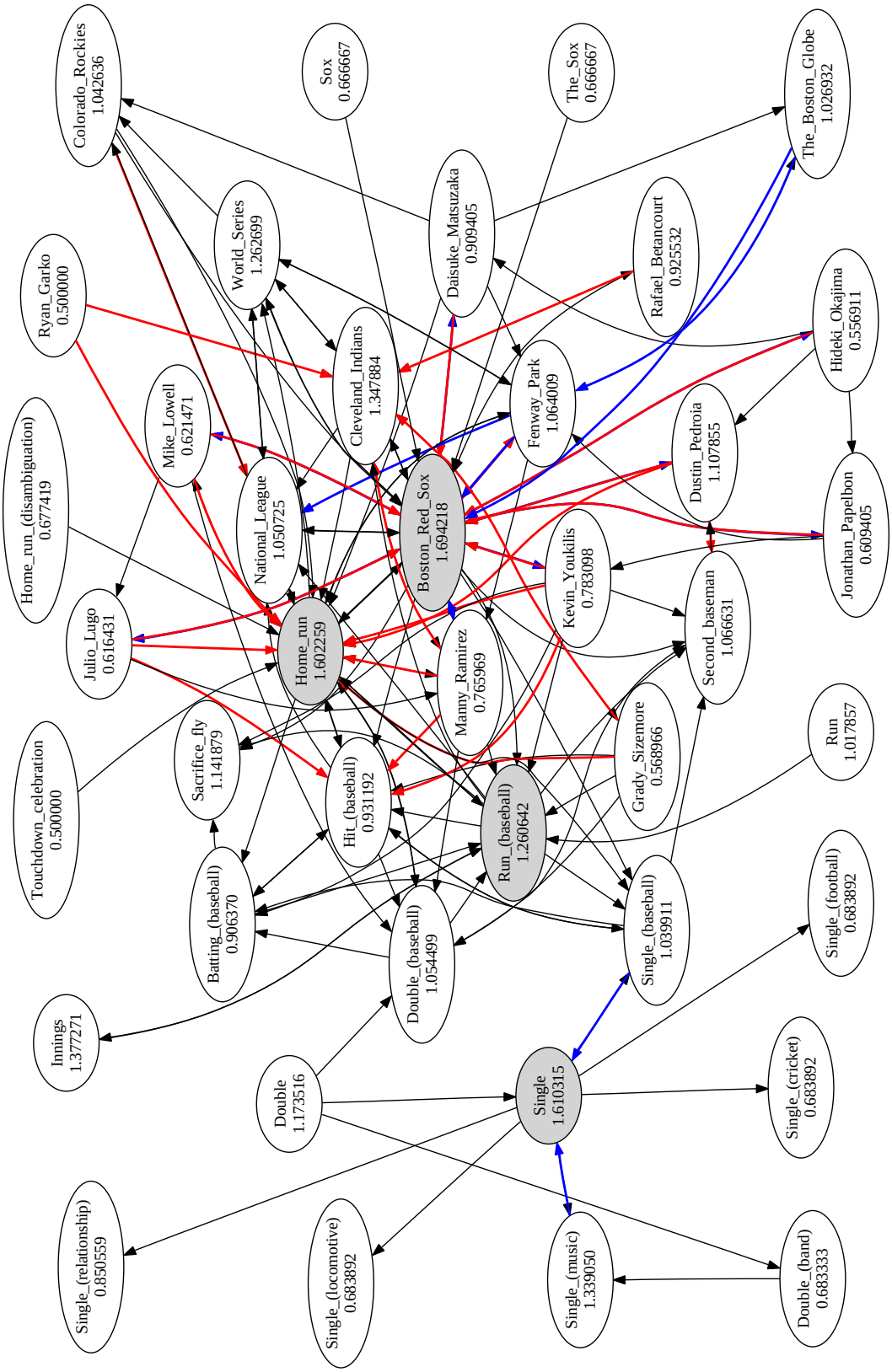

Figure 2. Thematic graph created for Baseball article from WikiNews

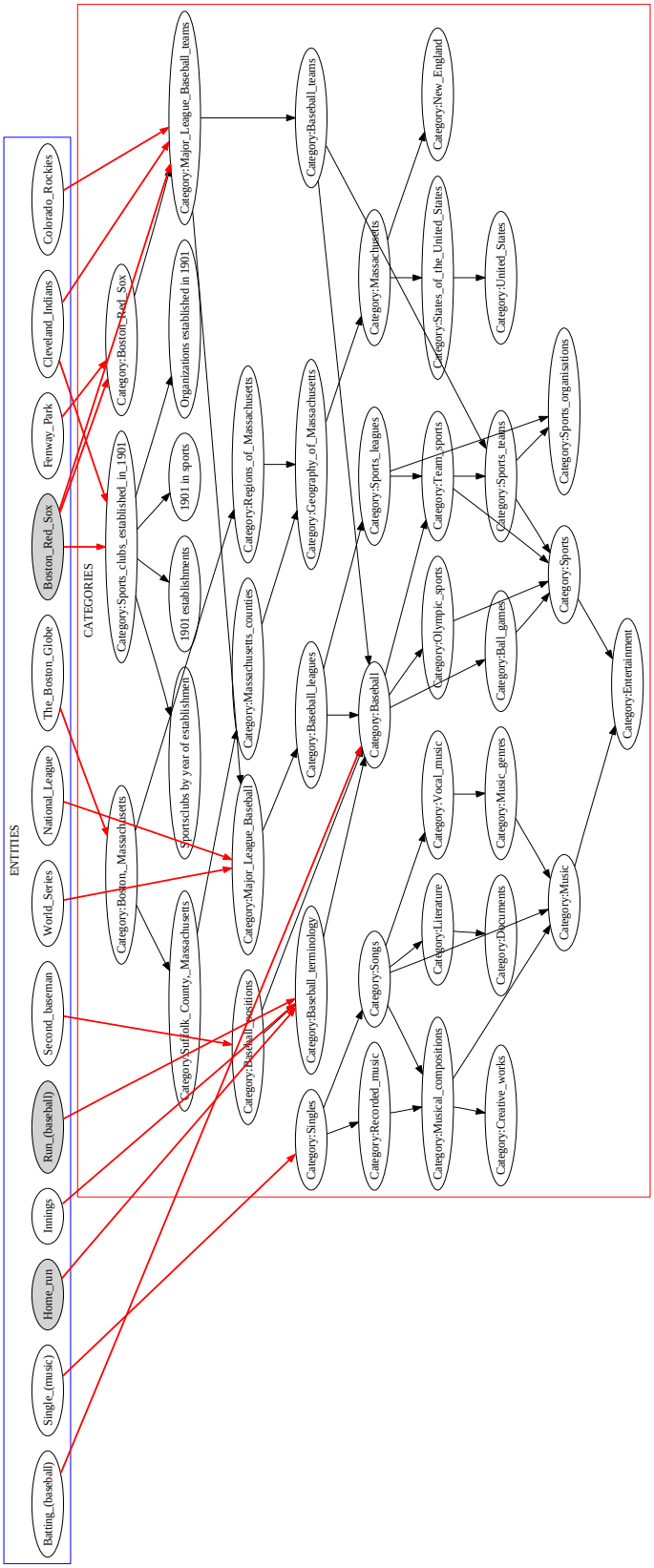

Figure 3. Partial category graph of Wikipedia categories for selected entities in thematic graph.

Optimizing single term queries using a personalized Markov random walk over the social graph

Maarten Clements[1,2], Arjen P. de Vries[2,1], Marcel J.T. Reinders[1]

[1]Information and Communication Theory Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands. [2]National Research Institute for Mathematics and Computer Science (CWI), The Netherlands.
E-mail: {m.clements, m.j.t.reinders}@tudelft.nl, Arjen@acm.nl

ABSTRACT

Social content systems contain enormous collections of unstructured user-generated content, annotated by the collaborative effort of regular Internet users. Tag-clouds have become popular interfaces that allow users to query the database by clicking relevant terms. However, these single click queries are often not expressive enough to effectively retrieve the desired content.

Using both rating and tagging information we have created a personalized retrieval model that effectively integrates the personal user preference in the content ranking. The soft clustering effect of our random walk model allows a smooth integration of concepts indirectly related to the target user and the query tag.

With collaborative annotations from a popular on-line book catalog, we show that our model outperforms standard tag-based retrieval. Both personalization and smoothing with closely related concepts significantly improve the content ranking. Our results indicate that individually created annotations are not semantically expressive enough to enable effective retrieval. Finally, we discuss the robustness of our model to well known linguistic problems like synonyms and homographs.

Keywords

Social Networks, Content Retrieval, Collaborative Tagging, Rating, Personalization

1. INTRODUCTION

In the last decade, the explosive use of digital budget cameras and integrated multimedia devices has resulted in an enormous increase in user-generated multimedia content like movieclips and pictures. On-line databases are actively used to store and share this content. Recently, the addition of social aspects in these databases has resulted in a large popularity increase. Millions of people use these *social content systems* to publish their creations or to be entertained by other people's contributions. Because the contributed data often does not carry a clear contextual description and there is no librarian to categorize the content, this has resulted in huge collections of unstructured data.

For future retrieval, many network users actively annotate the content using tags. Although most people use tagging to organize their own content collection, it has been shown that social tagging results in semantically descriptive annotations that can be used for content retrieval by the entire network [5, 11]. To initiate content retrieval, social tags are often shown in a *tag-cloud*, a visual depiction of tags in which the more popular tags are typeset in a larger font or more prominent color. Although there exist many different methods to draw these clouds [9], the relevance of a tag is often based on the global popularity of the tags in the entire network (e.g. popular tags in Last.fm[1]). In this way of navigation only a single popular word is used as a query, resulting in many retrieved documents. In traditional information retrieval (web-search engines), people often use multiple word queries in order to disambiguate their information need. To enable effective content ranking based on a single term, social content systems should be personalized to the user's preference.

In currently popular social content systems, there is a difference between *collaborative* tagging systems (e.g. CiteULike[2] and Del.icio.us[3]) and *individual* tagging systems (e.g. YouTube[4] and Flickr[5]). Many systems that allow user-generated content injection are individual tagging systems where only the injector of the content is able to assign the tags. In these systems, many people (who do not contribute any content) will not build up a profile of the tags they prefer. In collaborative tagging (CT), every user can tag any piece of content. In this way, users indicate which aspects of the content correspond to their personal interest. Also, in CT systems the aggregated tags of the network users create a relevance distribution for each content element. Furnas et al. already stated in 1987 that people often choose different terms to annotate content, resulting in low precision retrieval [4]. They argued that a theoretically optimal system would allow *unlimited aliasing* to describe the content. We advocate that collaborative tagging approaches unlimited aliasing and is therefore required to enable effective personalized content retrieval.

Besides tagging, the social aspects of networks stimulate people to share their opinion about the provided content. In many interfaces people can assess the quality of the content by giving a rating. With the introduction of ratings and tags in on-line databases, content annotation has shifted to subjective categorization. The combination of these two information sources creates a non-hierarchical database categorization based on both content quality and topic. Using ratings and tags, we create a graph of the network, resembling the actual relations in social content systems. We use a personalized random walk over this graph to evaluate the retrieval performance of single click queries.

[1]http://www.last.fm/tags
[2]http://www.citeulike.org
[3]http://del.icio.us
[4]http://www.youtube.com
[5]http://www.flickr.com

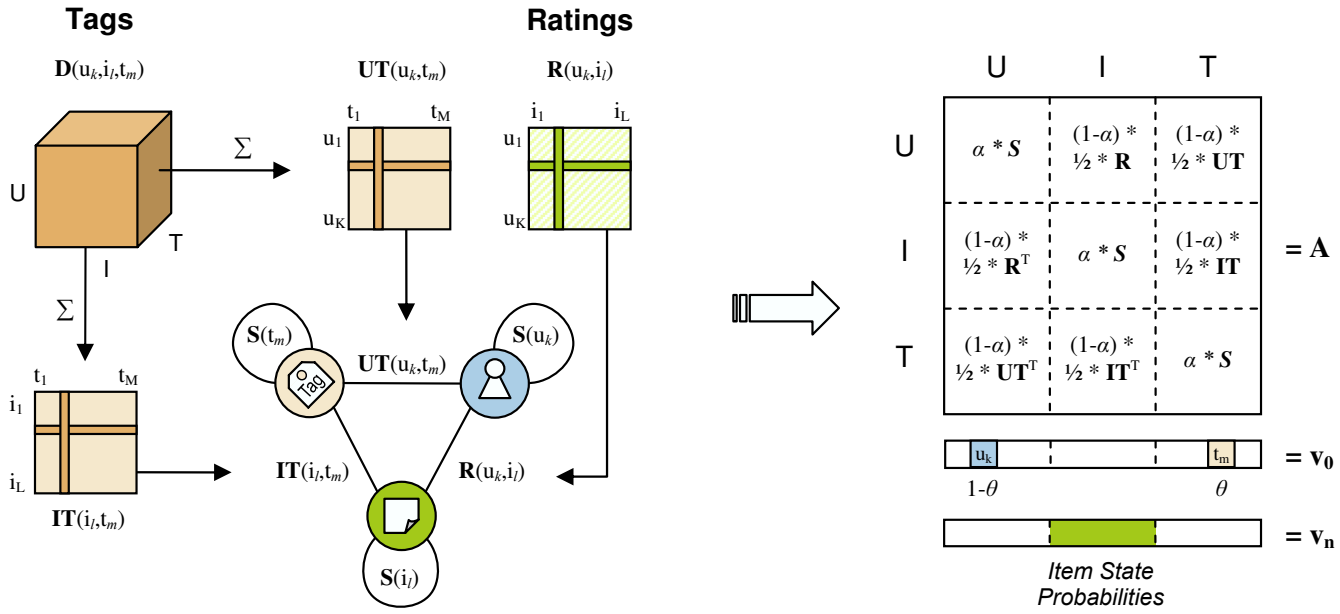

Figure 1: In our random walk model, the social content network is represented as a tripartite graph, containing users, items and tags as nodes. The edges between these entities are determined by rating ($\mathbf{R}$) or tag count ($\mathbf{UT}$, $\mathbf{IT}$). Self transitions ($\mathbf{S}$) allow the random walk to stay in the same node with a certain probability. Together, these edges constitute transition matrix $\mathbf{A}$. In the initial state vector $\mathbf{v}_0$, the index corresponding to the target user and the selected query tag are assigned with weights $1-\theta$ and θ. The result of the walk $\mathbf{v}_n$ contains the relevance probabilities of all three network elements. The model parameter α is used to tune the influence of self transitions.

2. PERSONALIZATION MODEL

For the relevance ranking of the content based on a selected tag we propose to use a random walk over the social graph, created by all rating and tagging actions. A random walk is a stochastic process in which the initial condition is known and the next state is given by a certain probability distribution. This distribution can be represented by the *transition matrix* $\mathbf{A}$, where $\mathbf{A}_{i,j}$ contains the probability of going from node i (at time n) to j (at time $n+1$):

$$\mathbf{A}_{i,j} = P(S_{n+1} = j | S_n = i) \tag{1}$$

The initial state can now be represented as a vector $\mathbf{v}_0$ (with $\sum(\mathbf{v}_0) = 1$), in which the query elements can be assigned. By multiplying the state vector with the transition matrix, we can find the state probabilities after one step in the graph ($\mathbf{v}_1$). Multi step probabilities can be found by repeating the multiplication $\mathbf{v}_{n+1} = \mathbf{v}_n\mathbf{A}$, or using the n-step transition matrix $\mathbf{v}_n = \mathbf{v}_0\mathbf{A}^n$. The number of steps taken in the random walk determines the influence of the initial state vector versus the background distribution. Under certain graph conditions, $\mathbf{v}$ will become stable (so that $\mathbf{v}_\infty = \mathbf{v}_\infty\mathbf{A}$) and in a completely connected graph it will contain the background probability of all nodes in the network (determined by the *volume* of connected paths).

Figure 1 shows how we create the transition matrix by combining rating and tagging information. If users, items and tags are seen as separate entities, the act of tagging creates a ternary relation between them [12]. These relations can be visualized in a 3D matrix $\mathbf{D}(u_k, i_l, t_m)$, where each position indicates if user u_k (with $k = \{1, \ldots, K\}$) tagged item i_l (with $l = \{1, \ldots, L\}$) with tag t_m (with $m = \{1, \ldots, M\}$).

Because even collaborative tagging systems are usually very sparse, we propose not to use the ternary relations directly, but sum over the 3 dimensions of $\mathbf{D}$ to obtain:

UT matrix: $\mathbf{UT}(u_k, t_m) = \sum_{l=1}^{l=L} \mathbf{D}(u_k, i_l, t_m)$, indicating how many items each user tagged with which tag.

IT matrix: $\mathbf{IT}(i_l, t_m) = \sum_{k=1}^{k=K} \mathbf{D}(u_k, i_l, t_m)$, indicating how many users tagged each item with which tag. In individual tagging systems, this will be a binary matrix.

UI matrix: $\mathbf{UI}(u_k, i_l) = \sum_{m=1}^{m=M} \mathbf{D}(u_k, i_l, t_m)$, indicating how many tags each user assigned to each item.

Earlier work on collaborative tagging systems proposed to create the social graph from the three projections of the ternary user-item-tag relation [12, 7]. Although the **UI** matrix contains interesting information about the users' tagging behavior, the relation between the number of tags assigned to an item and the preference of the user toward that item is unclear. Therefore, when modeling the users' preference, we replace the tag based User-Item matrix by the matrix based on the users' ratings. The rating matrix ($\mathbf{R}(u_k, i_l)$) contains the explicit users' preference for the available content, often expressed on a five or ten point scale.

Using the nonzero matrix values as edges, these three matrices ($\mathbf{UT}$, $\mathbf{IT}$ and $\mathbf{R}$) constitute a tripartite graph with users, items and tags as nodes. We include self-transitions that allow the walk to stay in place, which increases the influence of the initial state. The self transitions are represented in a diagonal matrix of ones $\mathbf{S} = \mathrm{diag}(1, \ldots, 1)$, so that the weight of the self transitions is equal for all nodes.

To reduce the influence of frequently occurring elements, we use TF-IDF weighing on the input matrices [15]. For example, the weighted User-Tag matrix is computed by:

$$\mathbf{UT}_{\text{TF-IDF}}(u_k, t_m) = \frac{\mathbf{UT}(u_k, t_m)}{\log \sum_{u=1}^{u=K} \mathrm{sgn}(\mathbf{UT}(u_k, t_m))} \tag{2}$$

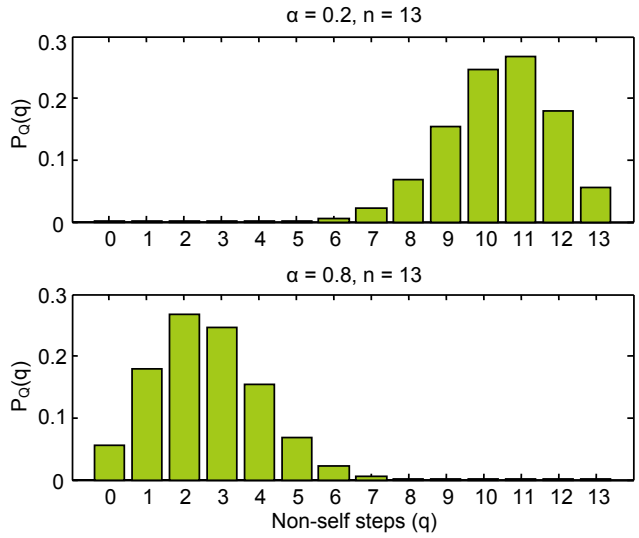

Figure 2: The PMF of the number of non-self steps after 13 steps through the social graph, for $\alpha = 0.2$ and $\alpha = 0.8$. Note that the number of non-self steps does not equal the distance to that starting node, because the walk might revisit earlier passed nodes.

where the sign function (sgn) sets all values > 0 to 1. Before combining the matrices we normalize them so that all rows sum to one.

We combine the **UT**, **IT** and **R** matrix in the transition matrix **A**, as shown in Figure 1. In this model $\alpha \in [0, 1]$ is the weight of the self transitions. Because the sub-matrices are normalized, the rows of **A** also sum to 1, so that they can be used as transition probabilities.

In the initial state vector, two starting points are assigned: $\mathbf{v}_0(u_k) = 1 - \theta$ and $\mathbf{v}_0(t_m) = \theta$ (where u_k is the target user and t_m indicates the selected tag). The parameter θ ($\theta \in [0, 1]$) determines the influence of the personal profile versus the query tag. The state probabilities after n steps are computed by repeating the multiplication of the state vector and the transition matrix **A**. After n steps, the content ranking is obtained by ordering the part of $\mathbf{v}_n$ that corresponds to content ($\mathbf{v}_n(K+1, \ldots, K+L)$) according to the state probabilities. This ranking will also contain the training data (i.e. the items already rated by the target user). We assume that a different user interface is used to browse previously seen content (the user's library), therefore we remove the training examples from the final ranking.

2.1 Self transition (α) and Walk length (n)

Depending on the number of steps in the random walk (n) the final ranking is mostly influenced by the starting points (target user and query tag) or the background distribution. The influence of the background after a certain number of steps is determined by the self-transition probability α. A large self transition probability allows the walk to stay in place (by taking many *self steps*), reinforcing the importance of the starting point, where a small value of α results in a walk that quickly converges to the stable background distribution.

Figure 2 shows the fraction of non-self steps for $\alpha = 0.2$ and $\alpha = 0.8$ at $n = 13$. Because all nodes have the same self transition probability, the total number of non-self steps (Q) after n steps through the social graph is a binomial random variable with the probability mass function (PMF):

$$P_Q(q) = \begin{cases} \binom{n}{q}\alpha^q(1-\alpha)^{n-q} & q = 0, \ldots, n \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Where $P_Q(q)$ is the probability of q non-self steps ($Q = q$).

The PMF shows that if a large value is chosen for α, most of the probability mass will stay close to the starting point and a long tail is created toward more distant nodes. With a small self transition probability the walk quickly moves away from the initial state. We choose a relatively high value of $\alpha = 0.8$ in our experiments to create a slow diffusion of the walk, because we expect to find the most relevant content close to the query, and we want enrich the ranking with a slow integration of more distantly related concepts.

2.2 Query weight (θ)

Most tag-based retrieval systems use the selected tag as query term and rank the content according to popularity ($P(i_l|t_m)$) or freshness ($P(i_l|time)$). Experience from the field of information retrieval has shown that a single term is often not semantically expressive enough to clearly define the user's content need. Our model enriches the query tag by integrating the users history in the search. In the initial state vector, both the query tag and the target user are assigned a value according to θ. The weight of this parameter determines the strength of the personalization. When θ is set to 0, the state probabilities only depend on the profile of the target user, so the predicted content ranking will not be relevant to the query, which closely resembles traditional collaborative filtering [14]. When $\theta = 1$ the state probabilities depend only on the selected query tag, so the result will not be personalized for u_k. If $0 < \theta < 1$ the model derives the probabilities, based on both the target user and the query.

3. DATA

3.1 LibraryThing

LibraryThing[6] is an on-line web service that allows users to create a catalog of the books they own or have read. A user can tag and rate all the books he adds to his personal library. The social aspects of this network give the user the opportunity to meet like-minded people and find new books that match his preference. The popularity of the system has resulted in a database that contains almost 3 million unique works, collaboratively added by more than 300,000 users. We are not aware of any other open network with this amount of collaborative tags ($\approx$ 30 million) and ratings ($\approx$ 3.5 million).

We have collected a trace from the LibraryThing network, containing 25,295 actively tagging users[7]. As expected in data organized by human-activity, we see that the number of books in the users' catalogs follows a power-law distribution [1] (see Fig. 3a). After pruning this data set we retain 7279 users that have all supplied both ratings and tags to at least 20 books. We remove books and tags that occur in less than 5 user profiles, resulting in 37,232 unique works and 10,559 unique tags. This pruned data set contains 2,056,487 UIT relations, resulting in a density of $7.2 * 10^{-7}$ (fraction of non empty cells in **D**). The derived **R**, **UT** and **IT** matrices have a density of respectively: $2.8 \cdot 10^{-3}$, $5.2 \cdot 10^{-3}$ and

[6] http://www.librarything.com
[7] Crawled in July 2007

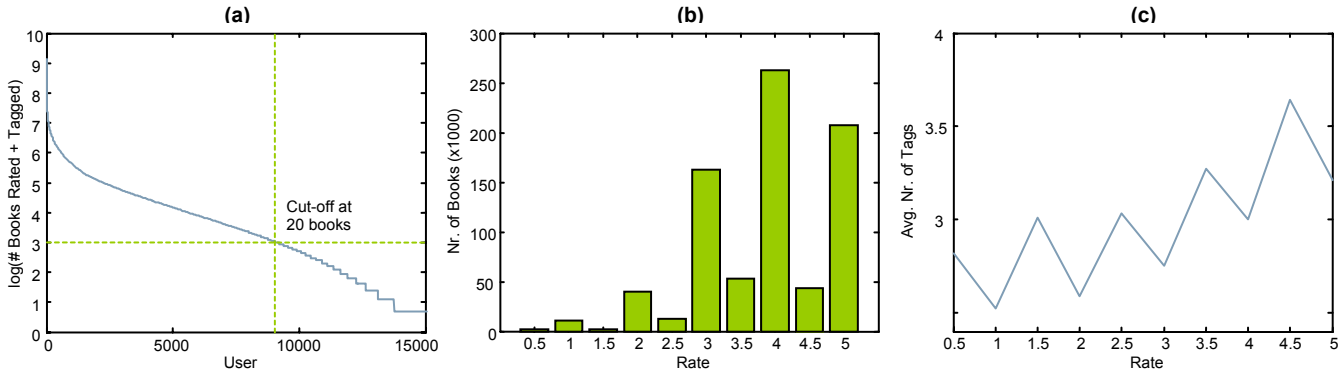

Figure 3: LibraryThing data statistics: a) The number of rated and tagged books stored in the users' catalogs, sorted by size. b) The distribution of rating occurrences in the pruned data set. c) The average number of tags assigned, given the rating.

$2.0 \cdot 10^{-3}$, and all show the power-law behavior common to social networks [6]. We expect that this data is comparable to collaboratively annotated movies, as books and movies comprise the same themes and storylines that can be categorized by tags.

The user interface of LibraryThing allows users to assign ratings on the scale from a half to five. Half ratings can be given by clicking a star twice. The distribution in Figure 3b shows that half ratings occur about 4 times less frequently than whole ratings. Figure 3c shows the relation between the rating and the number of tags given to an item. The upward trend shows that there is a slight correlation between these two variables. This graph also shows that books with half ratings tend to get more tags. This might indicate that the half ratings are used by people who put more effort in the categorization of their books.

4. EXPERIMENTAL SETUP

4.1 Data preparation

In order to estimate the performance of our model without overfitting to the data, we split the data in two equal parts (see Figure 4). Together with all the created annotations (ratings and tags), half of the users (3640 profiles) are put into the *training* set and the other half constitute the *test* set (*Step 1*). We now use the training set to optimize the model parameters by holding out 1/5 of the items of 1/5 of the training users (the validation set). We use our model to predict the held-out content (*Step 2*) using the tags assigned by the target user as query for the content he applied the tag on. Here we assume that the tags assigned to an item by the target user are the same words he would use as query to find the content. For each tag used by u_k we compute the NDCG measure discussed in the next section and compute the mean score over all validation users in the training set (*Step 3*, Figure 5 and 6).

The optimal model parameters derived from the training set are used to compute the performance on the test set, by holding again 1/5 of the user profiles of 1/5 of the users out, and computing the NDCG (*Step 4*). Finally we compare the results of our optimal model to the results achieved with conventional methods (*Step 5*, Table 1 and 2).

4.2 NDCG evaluation

To evaluate the predicted content ranking, we use the Normalized Discounted Cumulative Gain (NDCG) proposed by Järvelin and Kekäläinen [8].

In the predicted content ranking, the rank positions of the held-out validation ratings that correspond to a positive opinion $r \in \{3, 3.5, 4, 4.5, 5\}$ are assigned a value of respectively $G \in \{1, 2, 3, 4, 5\}$, called the *gain*. We do not normalize the rating profiles before assigning the gain, because we expect that the high offset in the ratings (See Figure 3b) is due to the fact that people tend to carefully select the books they read. As a result, people have read many more books they like than books they do not like.

In order to progressively reduce the gain of lower ranked test items, each position in the gain vector is discounted by the $^2\log$ of its index (where we first add 1 to the index, to ensure discounting for all rank positions > 0). The Discounted Cumulative Gain (DCG) now accumulates the values of the discounted gain vector:

$$\mathrm{DCG}[i] = \mathrm{DCG}[i-1] + G[i]/^2\log(i+1) \qquad (4)$$

The DCG vector is normalized to the optimal DCG vector. This optimal DCG is computed using a gain vector where all test rates are placed in the top of the ranking in descending order. Component by component division now gives us the NDCG vector in which each position contains a value in the range $[0, 1]$ indicating the level of perfectness of the ranking so far. We use the area below the NDCG curve as score to evaluate our rank prediction.

5. EXPERIMENTS

We will discuss the performance of our model on both collaborative and individual tagging systems. In all experiments we fix the self-transition probability (α) to 0.8.

5.1 Collaborative tagging

Parameter optimization To find the optimal model parameters and evaluate the sensitivity of the model we use our random walk to predict the left-out content of the training part of the LibraryThing data. Figure 5 shows the effect of the personalization at different walk lengths. The optimal NDCG is found at $\theta = 0.6$, which means that personalized retrieval gives a more accurate prediction than both completely personal and completely tag based queries.

We also find that the optimal number of steps is larger than one ($n_{optimal} = 13$), which means that the random walk improves a content ranking based on direct relations. Content that has not been tagged extensively will often miss

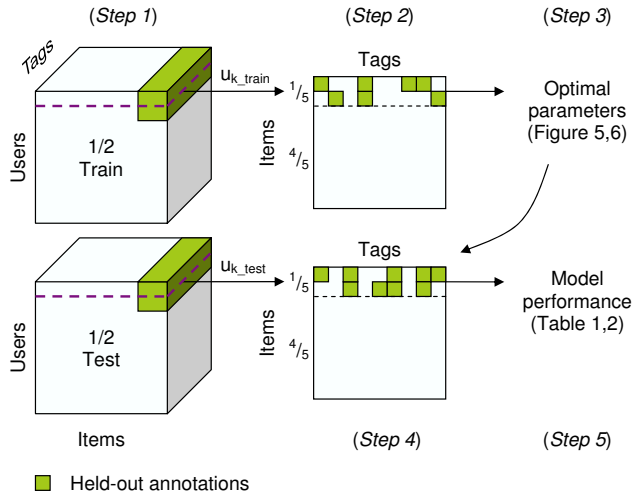

Figure 4: *Step 1*: Splitting of the $\mathbf{D}$ matrix, the $\mathbf{R}$ matrix is split accordingly. *Step 2,4*: A slice of the matrix contains a single user's items and tags. The tags used by that user are in turn used to predict the held-out content.

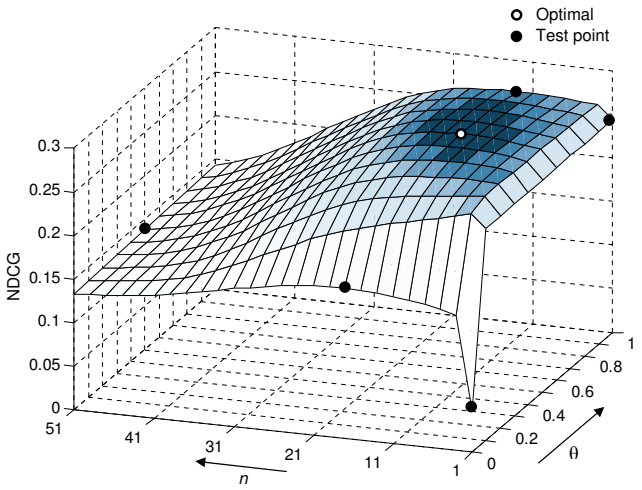

Figure 5: Optimization of the personalization influence θ and the walk length n. The optimal parameters and other test points are indicated with small circles. Note that the NDCG at $n = 1$ and $\theta = 0$ is equal to a random ranking, because a user has no direct link to potentially interesting content.

the terms used as a query by other people. The random walk can find these latent relations that are not explicitly present in the data.

The performance of the model is not very sensitive to small variations in both parameters. Because of the large self transition probability (α), the prediction slowly responds to variations in walk length. Only if the influence of the query tag is completely removed ($\theta \to 0$), the NDCG quickly drops to the global popularity score.

Test results To evaluate our model performance without overfitting to the data, we use a separate test as discussed in Section 4.1. We define two baseline methods: *Random*: The NDCG for a random ranking, and *Global Popularity*: The NDCG at $n = 51$ (We assume that the state vector is fully converged, so that $\mathbf{v}_{51} \approx \mathbf{v}_{\infty}$). We now compare four different model settings, derived from Figure 5. *Popularity Search*: Taking one step in our random walk model ($n = 1$) with $\theta = 1$ gives the ranking according to the number of times the tag was applied to the data. This parameter setting represents the tag browsing as implemented in many social content systems. *Random Walk Search*: Using the optimal number of steps at $\theta = 1$ represents the optimal performance with our model without personalization. Compared to popularity search, this method integrates more indirectly related concepts. *Recommendation*: When the model is completely personal ($\theta = 0$) the ranking will not depend on the query. Obviously this model setting gives lower performance, we see however that the performance is higher than the popularity ranking, indicating a strong coherence within the users' libraries. *Personalized*: The optimal parameter setting of our model.

The results show that both personalization and smoothing with indirectly related concepts by the random walk improve over traditional tag-based retrieval (See Table 1). Our *personalized* search model outperforms the usual *popularity search* by 11.1% and the random walk model without personalization (*Random walk search*) gives a gain of 9.0%.

Because much related work has used the social graph based on tagging information only [7, 10, 12], we have also optimized our model on the graph created with the **UI** matrix instead of the $\mathbf{R}$ matrix. We find that the performance is not significantly lower than the results with our model (Table 1, *No Rating*). This was already indicated by the correlation we found between the rating and the number of assigned tags (Figure 3c). We do however expect that in a data set with more negative opinions, the integration of the explicit preference information might give more performance gain over tag-based user-item relations, because it is impossible to assign a negative amount of tags.

Table 1: Collaborative tagging: Results on the test set

Model	θ	n	NDCG
Random	-	-	0.0466
Global Popularity	-	51	0.1574
Popularity Search	1	1	0.2378
RW Search	1	13	0.2591
Recommendation	0	17	0.1639
Personalized	0.6	13	0.2642
No Rating	0.6	13	0.2634

5.2 Individual tagging

To evaluate the benefit of a collaboratively annotated collection over individual tagging we adapt our data by removing all collaborative tags. For each book we randomly select one of its readers and keep only his tags to construct the graph. We use the tags that would be assigned by the other readers as their queries to retrieve the held-out content.

Parameter optimization The results on the training set are shown in Figure 6. We see that the optimal result is shifted to a higher value of θ, meaning that the required influence of personalization is much smaller. Also, a longer walk is needed to reach the optimal value. This can be explained by the fact that the reduced number of edges makes it harder to reach a large amount of relevant content in a small number of steps.

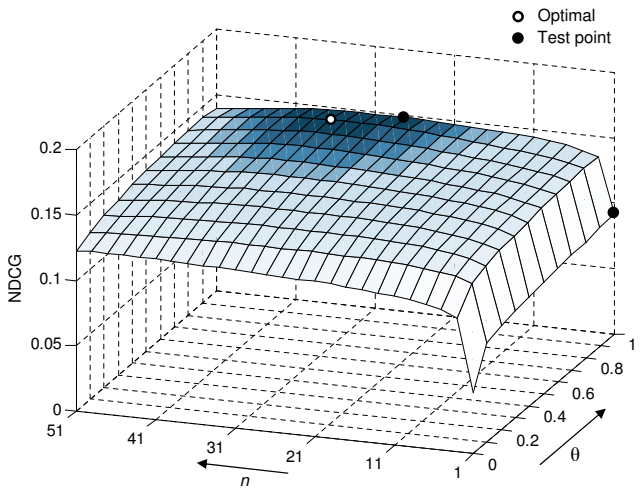

Figure 6: Optimization of the personalization influence θ in an individual tagging system.

Test results We show the results on the test set in Table 2. The NDCG gain of the personalized model over the non-personal model (*Random walk search*) is much smaller compared to the previously discussed results on collaboratively tagged data. In an individual tagging system the user profiles are very limited, because most users will have used only few tags or no tags at all. Therefore, the users' preference is mainly expressed by their given ratings, which appears to be a less informative representation with respect to focused retrieval.

Compared to results on collaborative tagging, the random walk on the individually tagged graph has more improvement over *popularity search* (61.9% improvement). This can be explained, as direct relations are extremely sparse and the random walk smoothly integrates more distantly related concepts. Popularity search performs even worse than a recommendation based on global popularity. Because users associate different terms with specific content, the retrieval model should take latent semantic relations into account, especially in individual tagging systems.

If we remove the rating information and create the social graph with the tag-based **UI** matrix, we see a significant performance drop (Table 2, *No Rating*). In individual tagging systems, the rating information is much more important because it allows people to create direct links with all content, instead of just the injected content.

Table 2: Individual tagging: Results on the test set

Model	θ	n	NDCG
Popularity Search	1	1	0.0926
RW Search	1	27	0.1475
Personalized	0.9	35	0.1499
No Rating	0.9	9	0.1036

6. DISCUSSION

6.1 Related work

A large part of the research on tagging systems has focused on the analysis of statistical patterns arising by the collaborative effort of network users. Golder and Huberman analyzed the structure of social bookmarking in Del.icio.us. They discovered recurring patterns of growth dynamics and identified various user tasks that result in different tagging behavior [5]. Halpin et al. extended this work by investigating the evolution of collaborative tagging patterns into stable distributions by computing the Kulback-Leibler divergence between different time points in Del.icio.us [6]. Marlow et al. showed that individual tagging systems evolve differently over time using data from the popular photo catalog Flickr [11]. Our results demonstrate that individual tagging also drastically reduces retrieval performance, which concurs with the *vocabulary problem* defined by Furnas et al., which states that people tend to use different terms to describe content [4].

Mika extended the bipartite ontology model used in traditional IR by directly integrating the network user in the graph [12]. The resulting tripartite graph gives more insight in the dynamics of social networks. Lambiotte and Ausloos used the same graph to visualize the network structure of Audioscrobbler[8] and CiteULike based on the projected matrices (**UT**, **IT** and **UI**) [10]. Hotho et al. also used the combination of the three binary graphs to apply a variation of adapted PageRank [13] on Del.icio.us data [7]. They only performed an empirical evaluation of their model, making it hard to compare. All these methods use the tag-based **UI** matrix, which does not precisely define the user-item relation. We showed that there is a slight correlation between preference and the number of tags assigned in LibraryThing (Figure 3c). Also, the performance between both information sources does not deviate significantly. However, in individual tagging systems, where most users are not able to apply tags to the content, the ratings provide essential information that can drastically improve content retrieval. We believe that when explicit user preference data is available, this information should be integrated in the social graph, especially in data with few tags or many negative ratings.

Our model is strongly related to the work of Craswell and Szummer [3]. They used a random walk on a query-image graph to retrieve more relevant images for each textual query. Instead of looking at the fully converged state vector ($\mathbf{v}_\infty$) that is used in PageRank related models, they also use the walk length as a model parameter. We have extended their model using the tripartite graph in which users, items and tags constitute the nodes. Because we directly integrate the network user in the model, the tasks that we describe are more focused on social interactions, which meets the desires of many current Internet users. Furthermore, in our model we always start the random walk from the target user, which makes the retrieval task personalized to each user's individual preferences.

6.2 Synonym and homograph robustness

Well known problems in tagging systems are synonyms and homographs. Synonyms are different words that share the same or closely related meaning. The problem in tagging systems arises, because there is no clear regulation on which words to use. If a piece of content has been tagged with a certain word and someone with a different background uses its synonym as a query, the content might not be found. The same problems arise when people use abbreviations, singular or plural words, word combinations and different languages. If a tag cloud is used to query a database, only a

[8]http://www.audioscrobbler.net/

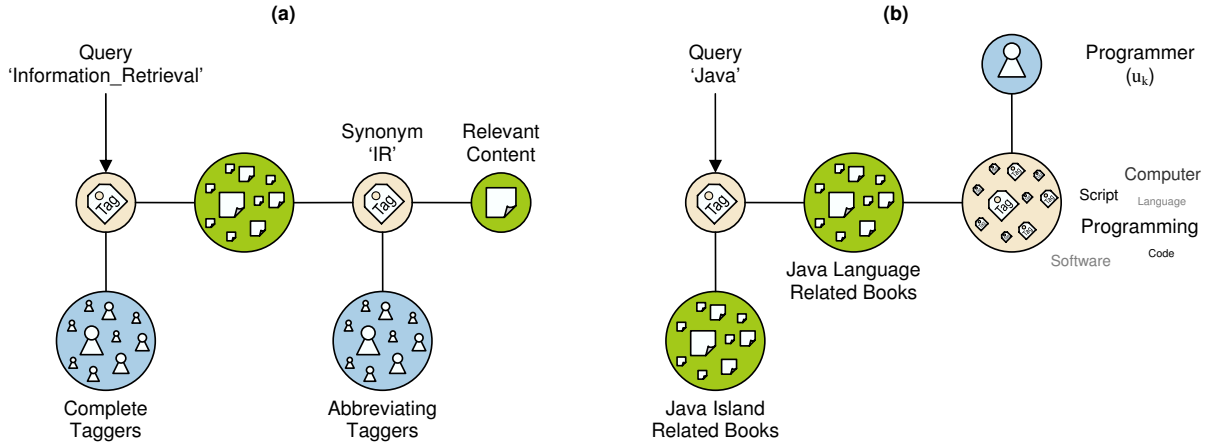

Figure 7: Our model is robust against both synonym and homograph problems: a) Synonyms or spelling differences (like 'Information_Retrieval' and the abbreviation 'IR') reinforce the content ranking because of the soft clustering created by the random walk model. b) Homographs like *Java* can be disambiguated using the target user's history.

single word is used as initial query resulting in sub-optimal retrieval performance.

Clustering methods have been proposed to group tags with strong lexical relations [2]. Clustering algorithms create binary relations between concepts although the natural similarity between words is a continuous relation. A random walk has shown to have a soft clustering effect that smoothly relates similar concepts before converging to the background probability [16]. Figure 7a shows that if enough users have tagged certain content, this soft clustering makes our model robust against synonymity problems, as synonym terms will be connected through a high volume of item connections.

Homographs are words that do not necessarily have the same pronunciation, but are written in exactly the same way. If a browsing user selects a homograph as query, the system will not know which denotation the user aimed for. In order to disambiguate the terms a user is looking for, our model integrates the information about the past behavior of that user. Because our random walk starts at both the query tag and the target user, the content that matches the target user's preference is more likely to be found first (see Figure 7b.).

7. REFERENCES

[1] A.-L. Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211.

[2] G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *WWW2006: Proceedings of the Collaborative Web Tagging Workshop*, Edinburgh, Scotland, 2006.

[3] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 239–246, New York, NY, USA, 2007. ACM Press.

[4] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971, November 1987.

[5] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32(2):198–208, April 2006.

[6] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 211–220, New York, NY, USA, 2007. ACM Press.

[7] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *ESWC 2006, LNCS 4011*, pages 411–426. Springer-Verlag Berlin Heidelberg, 2006.

[8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[9] O. Kaser and D. Lemire. Tag-cloud drawing: Algorithms for cloud visualization. In *Tagging and Metadata for Social Information Organization Workshop (WWW 2007)*, May 2007.

[10] R. Lambiotte and M. Ausloos. Collaborative tagging as a tripartite network. In *ICCS 2006, LNCS 3993*, pages 1114–1117. Springer-Verlag Berlin Heidelberg, May 2006.

[11] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM Press.

[12] P. Mika. Ontologies are us: A unified model of social networks and semantics. In Y. Gil, E. Motta, R. V. Benjamins, and M. Musen, editors, *ISWC 2005, LNCS 3729*, pages 522–536. Springer-Verlag Berlin Heidelberg, 2005.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[14] P. Resnick and H. R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, March 1997.

[15] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[16] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

Collaborative Annotation for Pseudo Relevance Feedback

Christina Lioma[1,2], Marie-Francine Moens[1], and Leif Azzopardi[2]

[1] Computing Science, Katholieke Universiteit Leuven, 3001, Belgium
[2] Computing Science, University of Glasgow, G12 8QQ, U.K.
{christina.lioma,sien.moens}@cs.kuleuven.be,
{xristina,leif}@dcs.gla.ac.uk

Abstract. We present a pseudo relevance feedback technique for information retrieval, which expands keyword queries with semantic annotation found in the freely available Del.icio.us collaborative tagging system. We hypothesise that collaborative tags represent semantic information that may render queries more informative, and hence enhance retrieval performance. Experiments with three different techniques of enriching queries with Del.icio.us tags, and also varying the number of tags used for expansion between 1-10, show small improvement in retrieval precision, over a baseline of short keyword queries.

1 Introduction

The field of Information Retrieval (IR) addresses the general problem of how to retrieve information, which is relevant to a user need, from a given repository of information, such as a document collection. A common example of IR systems is World Wide Web (Web) search engines, in which a short keyword query is used to generate a ranked list from a pre-indexed heterogeneous collection of documents. The matching between queries and documents is mostly term-based, i.e. the words within documents are used to describe the documents and to determine their relevance to a given query [25].

Often, the matching between queries and documents is enhanced by *relevance feedback*, which aims to render the initial query more informative, and resubmit it to the IR system, so that it can better match it to documents. There exist several ways for rendering queries more informative: (i) in *explicit relevance feedback* systems, users may expand their original query manually with potentially relevant terms suggested by the system [4, 6, 17, 21]; (ii) in *implicit relevance feedback* systems, logged user behaviour and/or search history can be used by the system to expand the original query automatically [2, 7, 8, 23]; (iii) in *pseudo relevance feedback* systems, term/document statistics can be used by the system to expand the original query automatically [16, 18]. Such term statistics can be extracted from documents already retrieved by the system (*local feedback*) [22], or from external sources of evidence, for instance Wikipedia entries [12] (*global feedback*) [11, 27].

In this paper we present a technique for expanding user queries with assumed relevant terms extracted from an external source of evidence, namely the Del.icio.us[3] collaborative annotation system. Del.icio.us is an online 'social tagging' system where users tag (= annotate), store and retrieve Web links. Given a query, Del.icio.us also suggests its most relevant tags. For example, given the query `holidays`, Del.icio.us suggests the related tags `travel, flights, calendar, hotels`[4]. We take advantage of this option, and expand a set of queries with their respective most related Del.icio.us tags. Our hypothesis is that such tags encode semantic information which may render the queries more informative and hence benefit retrieval performance.

We present three alternatives for selecting Del.icio.us tags: (i) on an individual term basis, (ii) on a phrase basis, (iii) on a whole query basis. Experimental evaluation of these techniques using the original (unexpanded) queries as baseline, on a standard Text REtrieval Conference (TREC[5]) dataset and with a robust model for matching documents to queries (Okapi's BM25 [15]) shows that our technique can improve retrieval precision for some but not all queries. This is a good starting point for further research into using collaborative annotation for IR. Given the free availability and increasing popularity (hence amount) of collaborative annotation, further research into incorporating this type of evidence in IR may be fruitful.

The remainder of this paper is organised as follows. Section 2 presents collaborative annotation systems and their use in IR. Section 3 presents our methodology for enriching queries with collaborative annotation. Section 4 presents and discusses our experiments. Section 5 summarises our findings and states intended future work.

2 Related studies

Broadly speaking, the underlying idea of semantic annotation is to identify interesting bits of metadata in documents (e.g. entities, relations, etc.). This type of annotation is becoming increasingly available online. For instance, the New York Times now uses rich headers metadata, while Reuters has launched the Open Calais[6] API for automatic semantic markup on HTML documents. Semantic annotation can be used in several ways to improve IR. For instance, knowledge of entities in text may be used to build sophisticated entity-based IR systems (sometimes referred to as *vertical search engines*). Another application is to automatically enrich textual content, for instance by inserting related links into raw text, as is done by the Inform[7] engine. Further applications include improving existing alert systems (e.g. RSS feeds), which are mostly based on keywords, and also incorporating on the fly text analysis into browsers. In brief,

[3] http://del.icio.us/
[4] Tags related to `holidays`, submitted to Del.icio.us on 14/02/2008.
[5] http://trec.nist.org/
[6] http://www.opencalais.com/
[7] http://www.inform.com/

semantic annotation is appealing because it can be seen as a way of enriching information (with more and structured data), which can result in improved processing; the question is whether this type of improved processing can result in improved system performance.

A type of semantic annotation is collaborative annotation, also known as *social tagging* or *distributed classification*, which refers to users creating and aggregating their own metadata. Collaborative annotation is a relatively new area (until recently largely absent from academic literature) but rapidly gaining ground on the Web.

The idea of asking users to annotate terms freely was initially developed by [5], who saw the process as a possible way of indexing particularly subjective forms of information where full-text searching was either not possible or not useful, such as multimedia or fiction objects. They developed the idea of aggregating users' indexing terms to create a generalised overall view of the resources, which today has been adapted by working systems, such as Del.icio.us, Flickr[8], a photo-sharing Web site where users upload, annotate and share photographs, CiteULike[9], a similar system but oriented towards scholarly writing and journal articles in particular, YouTube[10] and Last.fm[11], collaborative annotation services of multimedia resources (often user-authored). A potential disadvantage of human semantic annotation is inter-annotator disagreement or inconsistency, a result of allowing users to freely tag content. Early studies on human indexing also noted this as a problem [10, 19, 20].

The emergence of collaborative semantic annotation has stirred research in various directions, such as social issues surrounding tagging, growth and dynamics of social networks, cognitive processes behind tagging, and so on. The field of IR in particular has also shown interest in collaborative annotation: several commercial IR systems now include recommendation functionalities which are based on collaborative annotation, e.g. Amazon[12] uses collaborative annotation to suggest relevant products to online buyers. In addition, analogies between users - products in such recommender systems and queries - documents in IR systems are currently researched [13, 26].

3 Methodology for expanding queries with collaborative annotation

We present the steps taken in order to test the hypothesis that collaborative annotation includes semantic evidence, which can be used to enrich queries and hence enhance retrieval performance. Given a set of queries, for each query separately:

[8] http://www.flickr.com/
[9] http://www.citeulike.org/
[10] http://www.youtube.com/
[11] http://www.last.fm/
[12] http://www.amazon.com/

– **Step 1:** we submit it to Del.icio.us;
– **Step 2:** we use the θ most relevant tags returned by Del.icio.us to expand the original query;
– **Step 3:** we submit the expanded query to the IR system for retrieval.

We use three alternatives for **Step 1**:

– **term-based** alternative: we split the original query into individual terms, and submit each term separately to Del.icio.us. For example, original query=`foreign minorities, Germany`. Three separate queries submitted to Del.icio.us: (1) `foreign`, (2) `minorities`, (3) `Germany`.
– **phrase-based** alternative: we split the original query into phrases, and submit each phrase separately to Del.icio.us. For example, original query=`foreign minorities, Germany`. Two separate queries submitted to Del.icio.us : (1) `foreign minorities`, (2) `Germany`. We define phrases as comma-separated groups of terms.
– **query-based** alternative: we do not split the original query at all, but submit it as it is to Del.icio.us. For example, original query=`foreign minorities, Germany`=query submitted to Del.icio.us.

4 Evaluation

4.1 Experimental settings

The experimental aim is to test the hypothesis that expanding queries with collaborative semantic annotation can improve retrieval performance. We expect expanded queries to be more informative (which may increase early retrieval precision). The experimental setting is an IR system that matches documents to queries using an established retrieval model and unexpanded queries (baseline). To test our hypothesis, we expand queries with collaborative annotation from Del.icio.us, and compare retrieval performance to that of the baseline. We realise three rounds of experiments, one for each technique used in Step 1 to obtain related tags from Del.icio.us: (i) term-based, (ii) phrase-based, (iii) query-based.

We retrieve documents from the WT2G (2GB) collection, from the 1999, 2000 and 2001 Small Web tracks of the TREC Web Track (see Table 1), using topics 40-450 (see Table 4 in the Appendix). We use a Web dataset because it is more representative of real Web search. TREC queries usually contain a title, description, and narrative portion. The title contains few keywords; the description includes a brief description of the information need; the narrative contains a longer description of the information need. We experiment with short queries (title portion) only, because they are more representative of real user queries on the Web. We evaluate retrieval performance in terms of Precision at 10 (P10) and 20 (P20) retrieved documents.

We conduct experiments using the Terrier IR system [14]. Before retrieval, terms are tokenised on whitespace and punctuation marks, and lower-cased; stopwords are removed and terms are stemmed with the Porter stemmer. We

domain	size	#docs	#terms	#orig. query
Web	2GB	247,491	1,002,586	2.3

Table 1. The WT2G TREC dataset: domain = where it was crawled from; size = collection size; #docs = number of documents indexed; #terms = number of unique terms indexed; #orig. query = average length of original queries (unexpanded) measured in terms.

match documents to queries with the Okapi Best Match 25 (BM25) model [15]. BM25 includes certain parameters, which we set to default values. We use default values, instead of tuning these parameters, because our focus is to test our hypothesis, and not to optimise retrieval performance. If these parameters are optimised, retrieval performance may be further improved.

We expand the queries with θ terms suggested as relevant tags from Del.icio.us. We vary θ between 1-10, when possible (for some queries Del.icio.us offers < 10 relevant tags, see Table 5 in the Appendix). We treat Del.icio.us as a black box for suggesting relevant tags, i.e. we do not know how Del.icio.us estimates the relevance of the suggested tags, or whether these terms are ranked. We do not use queries numbered 407, 411, 414, 423, 427, 432, 438, 439, 440, 441, 442, 443, 449 because Del.icio.us does not suggest related tags for all of these query terms.

4.2 Experimental results

Figures 1 and 2 plot the number of collaborative tags used to expand queries (θ, x axis) against retrieval precision (y axis) for each of our three techniques used to select terms from Del.icio.us (term-based, phrase-based, query-based). We observe that overall, the term-based technique tends to perform worse, and that the phrase-based and query-based techniques perform approximately similarly, with the query-based technique giving slightly better results. This may be due to the compositional semantics of the whole query, which give a better representation of the user need than phrases or individual query terms. Figures 1 and 2 also show that precision seems to increase for lower θ values (= less expansion terms). In fact, the best performance is always associated either with $\theta = 0$, which corresponds to no query expansion (baseline), or with $\theta = 1$, which corresponds to the first relevant tag suggested by Del.icio.us (shown in Table 5 in the Appendix). Even though the estimation of relevance and the order of tag suggestion used by Del.icio.us is unknown to us, it seems that the most relevant tags come first, which may explain why $\theta = 1$ performs better.

Table 2 shows the P10 and P20 score of each query, separately for the original baseline queries (base) and for our three pseudo relevance feedback techniques (PRF_{term}, PRF_{phrase}, PRF_{query}), $\theta = 1$ (best θ, excluding the baseline). We observe that for about one third of the queries (13/36), the $\theta = 1$ term suggested by Del.icio.us is already a query term, e.g. term `Stirling` in query 447 (see Table 4). Out of the thirteen times that this happens, on one occasion P10 improves, and on two occasions P10 decreases, while for the remaining ten there is no change in performance. This is an interesting observation: the expansion

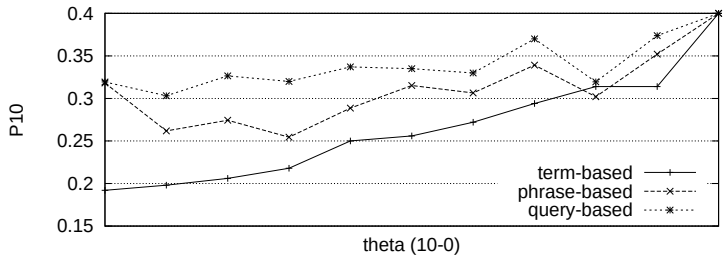

Fig. 1. Precision at 10 returned documents versus number of Del.icio.us terms used for query expansion. $\theta = 0$ is the baseline (no query expansion).

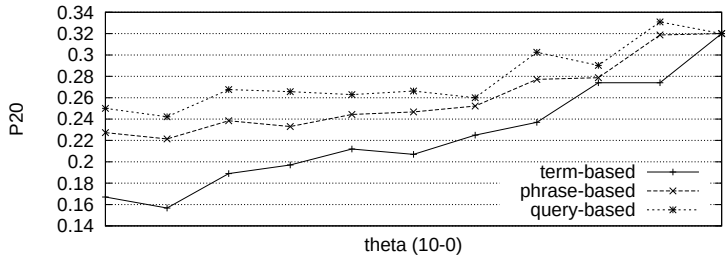

Fig. 2. Precision at 20 returned documents versus number of Del.icio.us terms used for query expansion. $\theta = 0$ is the baseline (no query expansion).

terms are providing redundant information, which explains why little change is observed overall in terms of performance. However, some interesting examples where performance did improve were on queries 404, 418, and 445, where the first Del.icio.us term seems helpful in emphasising the context of the query; e.g. in query 445 `women clergy`, the added term is `religion`, which is related to `clergy`; and in query 404 `Ireland peace talks`, the added term `activism` is again also related and on topic. While most of Del.icio.us terms seem to be related, not all are on topic. For instance, in query 404, Del.icio.us also suggests terms like `Iraq` and `Israel` where peace talks have been taking place, but are not on topic for the query. This suggests that the Del.icio.us tags might be better suited to aiding interaction, facilitating browsing or clustering data, instead of query expansion.

Table 3 compares baseline performance to the best performance marked by each of our three PRF techniques and also to a traditional PRF technique that expands queries with terms from the most relevant retrieved documents. For traditional PRF, we use the Bose Einstein 1 (Bo1) [1] term weighting model and add the 1-10 most relevant terms from the single top retrieved document. We observe that traditional PRF outperforms the baseline and our technique. This is expected, given that traditional PRF expands the query with 'local', 'weighted' relevant terms, while our technique expands the query with 'global', 'non-weighted' terms.

Results for best θ ($\theta = 1$)								
	Precision @ 10				Precision @ 20			
qid	base	PRF_{term}	PRF_{phrase}	PRF_{query}	base	PRF_{term}	PRF_{phrase}	PRF_{query}
401	0.30	0.20	**0.40**	0.30	0.40	0.20	0.25	0.20
402	0.50	**0.50**	**0.50**	**0.50**	0.50	**0.50**	**0.50**	**0.50**
403	0.90	**0.90**	**0.90**	**0.90**	0.90	**0.90**	**0.90**	**0.90**
404	0.50	**0.50**	0.40	**0.70**	0.55	0.50	0.30	**0.65**
405	0.00	**0.00**	**0.00**	**0.00**	0.05	0.00	**0.10**	**0.10**
406	0.20	0.00	0.00	0.00	0.20	0.05	0.15	0.15
408	0.60	**0.60**	**0.60**	**0.60**	0.35	**0.35**	**0.35**	**0.35**
409	0.30	0.20	0.20	**0.40**	0.25	**0.25**	**0.25**	**0.25**
410	0.80	0.70	0.50	0.50	0.75	0.65	**0.75**	**0.75**
412	0.70	**0.70**	**0.70**	**0.70**	0.70	**0.70**	**0.70**	**0.70**
413	0.00	**0.00**	**0.00**	**0.00**	0.10	0.00	0.00	0.00
415	0.40	0.30	**0.40**	**0.40**	0.35	0.20	**0.35**	**0.40**
416	0.60	0.50	0.30	0.30	0.50	**0.50**	**0.50**	**0.50**
417	0.40	**0.40**	**0.40**	**0.40**	0.30	**0.30**	**0.30**	**0.30**
418	0.40	0.60	**0.60**	**0.70**	0.30	**0.40**	**0.40**	**0.60**
419	0.20	0.10	**0.20**	**0.20**	0.20	**0.20**	0.15	**0.20**
420	0.60	**0.60**	**0.60**	**0.60**	0.35	**0.45**	**0.35**	**0.35**
421	0.60	0.10	0.20	0.20	0.35	0.20	0.20	0.20
422	0.00	**0.20**	**0.20**	**0.10**	0.10	**0.15**	**0.15**	0.05
424	0.20	**0.20**	**0.20**	**0.20**	0.20	**0.20**	**0.20**	**0.20**
425	0.40	0.20	**0.50**	**0.50**	0.35	**0.35**	**0.35**	**0.35**
426	0.10	0.00	0.00	**0.10**	0.05	**0.05**	**0.05**	**0.05**
428	0.10	**0.10**	**0.10**	**0.10**	0.05	**0.05**	**0.05**	**0.05**
429	0.40	**0.40**	0.30	0.30	0.30	0.25	**0.30**	**0.30**
431	0.20	**0.20**	0.10	0.10	0.25	**0.25**	**0.25**	**0.25**
433	1.00	0.40	0.40	0.60	0.55	0.25	0.25	0.55
434	1.00	0.90	0.90	**1.00**	0.65	0.60	0.60	**0.65**
435	0.10	**0.10**	**0.10**	**0.10**	0.05	**0.20**	**0.10**	**0.10**
436	0.00	**0.00**	**0.00**	**0.10**	0.10	**0.10**	0.05	0.05
437	0.10	**0.10**	**0.10**	**0.10**	0.05	**0.05**	**0.05**	**0.05**
444	1.00	**1.00**	**1.00**	**1.00**	1.00	**1.00**	**1.00**	**1.00**
445	0.00	**0.30**	**0.30**	**0.30**	0.15	**0.15**	**0.15**	**0.15**
446	0.40	**0.40**	**0.40**	**0.40**	0.30	**0.40**	**0.40**	**0.30**
447	1.00	0.80	0.80	0.80	0.60	0.55	0.55	0.55
448	0.00	**0.00**	**0.00**	**0.00**	0.00	**0.00**	**0.00**	**0.00**
450	0.40	0.00	**0.40**	**0.40**	0.30	0.10	**0.30**	**0.30**
Avg	0.40	0.34	0.35	0.38	0.32	0.31	**0.32**	**0.33**

Table 2. Precision at 10 and 20 relevant documents retrieved. Base = baseline (original queries). PRF_{term} = pseudo relevance feedback with Del.icio.us tags in response to individual terms. PRF_{phrase} = pseudo relevance feedback with Del.icio.us tags in response to phrases. PRF_{query} = pseudo relevance feedback with Del.icio.us tags in response to whole queries. Avg = average of all values. Bold = equal to or better than the baseline.

no PRF vs traditional PRF vs our PRF	P10	P20
baseline (no PRF)	0.40	0.30
traditional PRF (from relevant documents)	0.40	0.35
PRF_{term} (from Del.icio.us)	0.34	0.31
PRF_{phrase} (from Del.icio.us)	0.35	0.32
PRF_{query} (from Del.icio.us)	0.38	0.33

Table 3. Precision at 10 and 20 relevant documents retrieved. Baseline = original queries, no PRF. Traditional PRF = using the Bose Einstein 1 [1] term weighting model to expand queries with terms from the first relevant retrieved document. PRF_{term} = pseudo relevance feedback with Del.icio.us tags in response to individual terms. PRF_{phrase} = pseudo relevance feedback with Del.icio.us tags in response to phrases. PRF_{query} = pseudo relevance feedback with Del.icio.us tags in response to whole queries. For all PRF methods, we show values for the best number of expansion terms (θ between 1-10).

Overall, we observe that for most queries, our pseudo relevance feedback technique is either equal to or slightly better than the baseline. This seems to indicate that the contribution of the Del.icio.us semantic annotation is marginal. This may be due to our small dataset, or the techniques used for selecting terms for expansion without weighting them, but simply by considering them on a term- or phrase- basis. Perhaps more principled ways of selecting terms by weighting them, for instance by looking at their inverse document frequency in the collection, may benefit retrieval performance even more. The fact that there was no overall significant decrease of performance is encouraging, and indicates that this technique might be beneficial to retrieval on a selective basis, as has been shown with other forms of pseudo-relevance feedback [3].

5 Conclusion and future work

We presented a technique for pseudo relevance feedback, which expands queries with semantic annotation found in freely available collaborative tagging systems, and specifically Del.icio.us. We hypothesised that collaborative tags can represent semantic information that might be used to enrich queries, and hence enhance retrieval performance. We experimented with three different techniques of enriching queries with collaborative semantic annotation: (i) based on individual terms, (ii) based on phrases, and (iii) based on whole queries. We also experimented with the number of terms used for expansion, ranging it between 1-10. Out of the three techniques, the ones conveying context (phrase-based and query-based) behaved generally similarly; better performance was associated with the query-based technique and fewer expansion terms. Experiments with 36 Web queries showed no significant difference in retrieval performance between the original queries and the expanded queries. Some queries benefited from our technique, yet others did not; overall results are inconclusive. Collaborative semantic annotation seems to be broader than or quite general with respect to the user query, suggesting that perhaps better applications for it would be in

aiding user interaction, facilitating browsing and serendipitous search, or clustering documents, for instance. Further experimentation is needed in this direction, and particularly with regards to the selection of the most appropriate terms from the Del.icio.us related tags (e.g. by looking at their term statistics, or comparing their distribution in a general document collection to the distribution of query terms in the same collection, to identify discriminative terms).

In the future, we wish to experiment with larger datasets and more retrieval models (e.g. Inference Network Models [24] or Language Models [9], which allow for a straight-forward integration of evidence into the retrieval process and for weighting the effect of this integration), and with alternative ways of using collaborative semantic annotation to IR (e.g. to enrich documents, as opposed to queries only, a technique that might help to discriminate better between documents in a collection, and hence enhance retrieval performance).

References

1. Amati, G.: Probabilistic Models for Information Retrieval Based on Divergence from Randomness. PhD thesis, University of Glasgow, 2003.
2. Anick, S., P.: Using Terminological Feedback for Web Search Refinement: a Log-Based Study. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 2003) 88-95.
3. Cronen-Townsend, P., Zhou, Y., Croft, B.:: A Framework for Selective Query Expansion. In: ACM Conference on Information and Knowledge Management (CIKM 2004) 236-237.
4. Harman, D.: Towards Interactive Query Expansion. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 1988) 321-331.
5. Hidderley, R., Rafferty, P.: Democrating Indexing: an Approach to the Retrieval of Fiction. Information Services & Use, 17(23). (1997) 101-109
6. Koenemann, J., Belkin, N.: A Case for Interaction: a Study of Interactive Information Retrieval and Effectiveness. In: Conference on Human Factors in Computing Systems (SIGCHI 1996) 205-212.
7. Komlodi, A., Soergel, D., Marchionini, G.: Search Histories for User Support in User Interfaces. JASIS, 57(6). (2006) 803-807
8. Lau, T, Horvitz, E.: Patterns of Search: Analyzing and Modeling Web Query Refinement. Proceedings of the 7th International Conference on User Modelling. (1999) 119-128.
9. Lavrenko, V., Croft, B.: Relevance-Based Language Models. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 1999) 120-127
10. Lewis, D. D., Sparck Jones, K.: Natural Language Processing for Information Retrieval. In: Communcations of the ACM, (1996) 39(1):92-101.
11. Mandala, R., Tokunaga, T., Tanaka, H.: Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 1999) 191-197
12. Milne, D., Witten, I. H., Nichols, D. M.: A Knowledge-Based Search Engine Powered by Wikipedia. In: CIKM, (2007) 445-454.
13. Mishne, G.: AutoTag: a Collaborative Approach to Automated Tag Assignment for Weblog Posts. In: Proceedings of the 15th international conference on World Wide Web, (2006) 253-254.

14. Ounis, I., Lioma, C., Macdonald, C., Plachouras, V.: Research Directions in Terrier: A Search Engine for Advanced Retrieval on the Web. In: Ricardo Baeza-Yates et al. (Eds.) Novatica/UPGRADE Special Issue on Web Information Access, invited paper, 2007.
15. Robertson, S., Walker, S.: Some Simple Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 1994) 232-241
16. Rocchio, J.: Relevance Feedback in Information Retrieval. In: The SMART Retrieval System, (1971) 313-323.
17. Ruthven, I.: Human Interaction: re-examining the potential effectiveness of interactive query expansion. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 2003) 213-220.
18. Salton, G, Buckley, C.: Improving Retrieval Performance by Relevance Feedback. In: JASIS (1990), 41:288-297.
19. Salton, G, McGill, M. J.: Introduction to Modern Information Retrieval. MacGraw-Hill, New York (1983).
20. Soergel, D.: Indexing and Retrieval Performance: The Logical Evidence. In: JASIS (1994), 45(*):588-599.
21. Spink, A.: Term relevance feedback and query expansion: relation to design. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 1994) 81-90.
22. Tan, B., Atulya, V., Fang, H., Zhai, C. X.: Term Feedback for Information Retrieval with Language Models. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 2007) 263-270.
23. Teevan, J., Adar, E., Jones, R., Potts, M. A. S.: Information Re-Retrieval: Repeat Queries in Yahoo's Logs. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 2007) 151-158.
24. Turtle, H. R., Croft, W. B.: Inference Networks for Document Retrieval. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 1990) 1-24.
25. van Rijsbergen, C., J.: Information Retrieval. Butterworths, London (1979)
26. Wang, J., De Vries, A., Reinders, M. j. T.: Unified Relevance Models for Rating Prediction in Collaborative Filtering. In: ACM Transactions on Information Systems, in press.
27. Xu, J., Croft, B.: Query Expansion using Local and Global Document Analysis. In: ACM Conference on Research and Development in Information Retrieval (SIGIR 1996) 4-11.

qno	TREC query title	TREC query description
401	foreign minorities, Germany	**language, cultural**, differences, impede, integration, foreign, minorities, Germany
402	behavioral genetics	happening, field, **behavioral, genetics**, study, relative, influence, **genetic** environmental, factors, individual's, **behavior**, personality
403	osteoporosis	find, information, effects, dietary, intakes, potassium. magnesium, fruits, vegetables, determinants bone, mineral, density, elderly, men, women, preventing, **osteoporosis**, bone, decay
404	Ireland, peace talks	often, peace, talks, ireland, delayed, disrupted, result, acts, violence
405	cosmic events	unexpected, unexplained, cosmic, events, celestial, phenomena, radiation, supernova, outbursts, new, comets, detected
406	Parkinson's disease	being, done, treat, symptoms, **parkinson's, disease**, keep, patient, functional, long, possible
408	tropical storms	**tropical, storms, hurricanes**, typhoons, caused, significant, property, damage, loss, life
409	legal, Pan Am, 103	legal, actions, resulted, destruction, pan am, flight, 103, lockerbie, scotland, december 21 1988
410	Schengen agreement	involved, **schengen**, agreement, eliminate, border, controls, western, europe, hope, accomplish
412	airport security	**security**, measures, effect, proposed, go, effect, **airports**
413	steel production	new, methods, producing, **steel**
415	drugs, Golden Triangle	**drugs**, known, trafficking, golden, triangle, area, burna, thailand, laos, meet
416	Three Gorges Project	status, **three, gorges**, project
417	creativity	find, ways, measuring, **creativity**
418	quilts, income	ways, quilts, used, generate, income
419	recycle, automobile tires	new, uses, developed, old, automobile, tires, means, tire, recycling
420	carbon monoxide poisoning	widespread, **carbon**, monoxide, global, scale
421	industrial waste disposal	disposal, industrial, **waste**, being, accomplished, industrial, management, world
422	art, stolen, forged	incidents, stolen, forged, art
424	suicides	give, examples, alleged, **suicides**, aroused, suspicion, **death**, actually, being, murder
425	counterfeiting money	counterfeiting, **money**, being, done, modern, times
426	law enforcement, dogs	provide, information, use, **dogs**, worldwide, **law**, enforcement, purposes
428	declining birth rates	countries, U.S., china, declining, birth, rate
429	Legionnaires' disease	identify, outbreaks, legionnaires', disease
431	robotic technology	latest, developments, **robotic, technology**
433	Greek, philosophy, stoicism	contemporary, interest, **greek, philosophy, stoicism**
434	Estonia, economy	state, **economy, estonia**
435	curbing population growth	measures, taken, worldwide, countries, effective, curbing, **population**, growth
436	railway accidents	causes, railway, **accidents**, world
437	deregulation, gas, electric	experience, residential, utility, customers, following, deregulation, gas, electric
444	supercritical fluids	potential, uses, **supercritical, fluids**, environmental, protection, measure
445	women clergy	countries, United, states, considering, approved, **women, clergy**, persons
446	tourists, violence	tourists, likely, subjected, acts, **violence**, causing, bodily, harm, death
447	Stirling engine	new, developments, applications, **stirling, engine**
448	ship losses	identify, instances, weather, main, contributing, factor, loss, ship, sea
450	King Hussein, peace	significant, figure, years, late, jordanian, king, hussein, furthering, peace, middle, east

Table 4. Original queries used in the experiments; qno = query number. Titles and descriptions as provided by TREC. Bold = description terms also suggested by Del.icio.us as relevant (see Table 5).

qno	original query	expansion terms (most common del.icio.us annotation)
401	foreign minorities, Germany	culture, language, languages, linguistics, reference
402	behavioral genetics	genetics, science, psychology, evolution, research, biology, behavior, sociology, economics, philosophy
403	osteoporosis	osteoporosis, health, nutrition, medical, food, medicine, calcium, arthritis, fitness
404	Ireland, peace talks	activism, audio, campaign, charity, environment, eu, green, Iraq, Israel, media
405	cosmic events	science, astronomy, space, news, interesting, physics, article, daily, cool, future
406	Parkinson's disease	health, parkinsons, science, parkinson's, brain, research, disease, medicine, parkinson, politics
408	tropical storms	weather, hurricane, hurricanes, news, maps, science, storm, reference, tropical, noaa
409	legal, Pan Am, 103	reference, dictionary, google, language, map, maps, thesaurus, travel, visualization
410	Schengen agreement	schengen, eu, politics, international, travel, visa, wiki, wikipedia
412	airport security	security, travel, airport, politics, terrorism, wifi, mac, tsa, wireless, osx
413	steel production	design, business, art, steel, diy, reference, technology, tools, engineering, hardware
415	drugs, Golden Triangle	drugs, ajax, asia, homepage, news, police, politics, portal, rss, strange
416	Three Gorges Project	china, environment, energy, dam, 3gorges, bbc, gorges, news, photos, three
417	creativity	creativity, design, inspiration, productivity, art, blog, lifehacks, innovation, writing, business
418	quilts, income	art, crafts, design, handmade, shopping
419	recycle, automobile tires	architecture, cool, destruction, environment, fun, green, shredding, sustainability, sustainable, video
420	carbon monoxide poisoning	suicide, carbon, crossover, health, poisoning
421	industrial waste disposal	recycling, recycle, waste, market, plastic, environment, management, scrap, photography, art
422	art, stolen, forged	audio, free, manifesto, music, opensource, pandora, radio, software, technology, web2.0
424	suicides	suicide, funny, humor, comics, death, war, bunny, politics, iraq, news
425	counterfeiting money	money, security, politics, privacy, counterfeit, currency, crime, economics, economy, printer
426	law enforcement, dogs	law, police, accessibility, activism, ada, censorship, disability, dog, doghouse, dogs
428	declining birth rates	articles, parenting
429	Legionnaires' disease	uk
431	robotic technology	technology, robotics, robots, robot, science, art, video, design, diy, electronics
433	Greek, philosophy, stoicism	philosophy, stoicism, classics, religion, books, epictetus, ethics, greek, history, jesus
434	Estonia, economy	estonia, politics, economics, statistics, economy, culture, europe, freedom, tax, bots
435	curbing population growth	development, environment, health, population, poverty
436	railway accidents	accident, activism, alcohol, article, design, disasters, madd, politics, risk, technology
437	deregulation, gas, electric	energy, engineering, engineers, organization, organizations
444	supercritical fluids	chemistry, fluids, nature, news, physics, science, supercritical, water
445	women clergy	religion, islam, feminism, politics, christianity, philosophy, women
446	tourists, violence	blog, china, crime, culture, drugs, egypt, humor, literature, police, violence
447	Stirling engine	stirling, engine, energy, science, solar, diy, power, technology, environment, howto
448	ship losses	music, riaa, business, apple, businessmodel, filesharing, future, hardware, lies, mac
450	King Hussein, peace	animation, bush, engine, flash, funny, google, israel, politics, search, searchengine

Table 5. Queries used in the experiments and their respective relevant terms in the order suggested by Del.ico.us and used in our query-based technique; qno = query number.

Web Search Disambiguation by Collaborative Tagging

Ching-man Au Yeung, Nicholas Gibbins, and Nigel Shadbolt

Intelligence, Agents, Multimedia Group,
School of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK
{cmay06r,nmg,nrs}@ecs.soton.ac.uk

Abstract. Existing Web search engines such as Google mostly adopt a keyword-based approach, which matches the keywords in a query submitted by a user with the keywords characterising the indexed Web documents, and is quite successful in general in helping users locate useful documents. However, when the keyword submitted by the user is ambiguous, the search result usually consists of documents related to various meanings of the keyword, in which probably only one of them is interesting to the user. In this paper we attempt to provide a solution to this problem by using the semantics extracted from collaborative tagging in the social bookmarking site del.icio.us. For an ambiguous word, we extract sets of tags which are related to it in different contexts by performing a community-discovery algorithm on folksonomy networks. The sets of tags are then used to disambiguate search results returned by del.icio.us and Google. Experimental results show that our method is able to disambiguate the documents returned by the two systems with high precision.

1 Introduction

The amount of information on the World Wide Web is huge and keeps increasing as people from all over the world continue to contribute to this information network. It was estimated that there were already over 11.5 billion Web pages on the Web as of the end of January 2005 [4]. Such rapid growth has made the retrieval of information that is relevant to the needs of a user very difficult. While search engines help ease the problem by indexing the Web and returning search results based on ranking algorithms such as the PageRank [2] algorithm, in many situations the results returned are not as useful as the users have expected.

An obvious example of such situations is when a keyword with multiple meanings is used to query the search engines. Very often the retrieved documents are relevant to multiple meanings of the keyword, but the user is probably only interested in one of the meanings or one of the contexts in which the keyword is used. For example, when a user queries Google with the keyword *bridge*, he might be presented with Web pages about bridge as a kind of card game, as a

design pattern in programming, or as a physical structure built across a river. The user who is only interested in bridge as a kind of card game will have to scan through the list of returned documents and single out those which are really relevant.

In recent years, collaborative tagging systems such as del.icio.us and Flickr have become very popular among Web users as a means of organising their favourite Web resources.[1] In these systems, users are allowed to choose any words they like as tags to describe Web resources, resulting in a user-generated classification scheme now commonly known as a *folksonomy* [20]. Not only does a folksonomy provide metadata of Web resources in the form of tags, it also provides a lot of information on the relations between different tags when they are used together. We have shown [1] that by performing clustering on documents tagged in a folksonomy, it is possible to extract the sets of tags related to the different contexts in which an ambiguous tag is used.

In this paper, we discuss how such implicit semantics extracted from a folksonomy can be utilised to enhance Web search. We propose a method to disambiguate Web search results by classifying returned documents into different contexts in which an ambiguous keyword is used. Evaluation is performed by applying the method on search results returned by del.icio.us and Google.

The remaining of this paper is structured as follows. The next section presents an example which motivates this research. In Section 3, we describe our method for automatically extracting the different meanings of an ambiguous tag from a folksonomy. In Section 4, we describe how we apply the results of tag meaning disambiguation on Web search disambiguation. Section 5 presents the experimental results. Finally, we mention some related work in Section 6 and give conclusions and future research directions in Section 7.

2 Motivating Example

It is very common for a user of a Web search engine to find that the search results are not as useful as expected. This is particularly true when the keywords used in the query represent different concepts when used in different contexts. In such cases the users have to go through the list of returned documents and single out those documents that are relevant to their needs.

Consider the following example of searching for information about the type of card game called Bridge. Table 1 lists the top ten pages returned by Google UK when *bridge* is used as the query string. While the first and the third pages returned are about the card game, the search results actually consist of pages about other meanings of the word *bridge*. For example, the second item is a page from Wikipedia describing bridges as architectural structures, and the sixth item is a page which contains travel information about the Golden Gate Bridge. There are also pages (e.g. 7th and 10th) which involve organisations or projects with the name 'Bridge' but are by no means related to any commonly used meanings of the word.

[1] http://del.icio.us/, http://www.flickr.com

1	Contract bridge - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Contract_bridge
2	Bridge - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Bridge
3	Play bridge card game online http://www.bridgeclublive.com/
4	Bridge Travel http://direct.bridge-travel.co.uk/
5	River Kwai Bridge Travel http://www.riverkwaibridge.com/
6	Golden Gate Bridge Guide — Attraction Travel Guide http://www.worldtouristattractions.travel-guides.com/attraction/170/ attraction_guide/North-America/Golden-Gate-Bridge.html
7	Bridge - Mainstreaming Gender Equality http://www.bridge.ids.ac.uk/
8	Bridge to Reuters http://www.bridge.com/
9	The Bridge SE1 - London venue for parties, gigs, films, conference http://www.thebridgese1.co.uk/
10	BRIDGE (Building Radio Frequency IDentification for the Global Environment) http:// www.bridge-project.eu/

Table 1. The top ten pages returned by Google UK when '*bridge*' is used as a query string.

Two major problems can be observed in this example. Firstly, extra effort is required for the user to go through the list and select those results which are useful. Secondly, the presence of pages which are irrelevant to the user's need reduces the number of relevant pages that can be presented to the user at one time, especially when users tend to only inspect the first set of items returned [8, 18]. Although in some search engines terms which are commonly used together with the keyword are suggested to the users for refining the search results, it will be much more efficient if the search engine is able to classify the pages into different categories which correspond to different meanings of the keyword before presenting the results to the user. To tackle these problems, we propose a method for Web search disambiguation by using the semantics extracted from a folksonomy.

3 Tag Meaning Disambiguation

Our first step to Web search disambiguation involves obtaining the different meanings of an ambiguous word from a folksonomy.[2] We have shown that, for

[2] Since a word is referred to as a tag, a keyword or a term depending on the context in which it is being mentioned, we will use these terms interchangeably in the rest of this paper.

an ambiguous tag in a folksonomy, documents which are relevant to the same meaning of the tag tend to be grouped together [1]. This suggests that clustering algorithms can be applied to extract groups of documents which correspond to different meanings of the tag. Our target is to extract sets of tags which constitute different contexts in which an ambiguous tag is used. The proposed process for tag meaning disambiguation is described as follows.

A folksonomy is generally considered to consist of at least three sets of elements [13, 21], namely users, tags, and documents. Formally, we define a folksonomy as a tuple $\mathbf{F} = (U, T, D, A)$, where U is a set of users, T is a set of tags, D is a set of Web documents, and $A \subseteq U \times T \times D$ is a set of annotations. When we want to understand the different meanings of an ambiguous tag t, only a subset of the folksonomy involving the tag is required. This can be obtained by extracting the bipartite graph UD_t by restricting $\mathbf{F}$ to t:

$$UD_t = \langle U \cup D, E_{ud} \rangle, E_{ud} = \{(u, d) | (u, t, d) \in A\}$$

This graph can be represented in matrix form, which we denote as $\mathbf{Y} = \{y_{ij}\}$, $y_{ij} = 1$ if there is an edge connecting user u_i and document d_j, and $y_{ij} = 0$ otherwise. We further fold this bipartite graph into a one-mode network of documents by performing matrix multiplication, obtaining $\mathbf{C} = \mathbf{Y}'\mathbf{Y}$. In this one mode network, an edge is weighed by the number of users who have assigned tag t to the documents represented by the vertices on the two ends of the edge.

From this network of documents, we can extract groups of documents where each group corresponds to a single meaning of the tag t. This can be done by applying clustering algorithms to the network represented by $\mathbf{C}$. We adopt the fast greedy algorithm for community discovery in networks proposed in [15], which optimises modularity [16] by connecting the two vertices at each step which result in the largest increase (or smallest decrease) of modularity. If D_t is the set of documents which are assigned the tag t, the result of the clustering process is a set of sets of documents: $\mathbf{X}_t = \{X_{t,1}, X_{t,2}, ..., X_{t,m}\}$ where $X_{t,1} \cup X_{t,2} \cup \cdots \cup X_{t,m} = D_t$. Finally, for each set $X_{t,i}$, we obtain a set $T_{t,i}$ of the top 10 tags which are used most frequently by the users on the documents in the set.

While each of these sets of tags is likely to be related to a single meaning of the ambiguous tag t, it is possible that two or more of these sets are related to the same meaning. To eliminate the redundancy in the result we combine two sets of tags if there is significant overlap between the two with the help of the following function:

$$overlap(T_{t,i}, T_{t,j}) = \frac{|T_{t,i} \cap T_{t,j}|}{|T_{t,i} \cup T_{t,j}|} \quad (1)$$

We introduce a threshold α, and merge the two sets of documents $X_{t,i}$ and $X_{t,j}$ when $overlap(T_{t,i}, T_{t,j}) \geq \alpha$. The top 10 tags with the highest frequencies are extracted to form a new set. Hence, the final result of this tag meaning disambiguation process is a set of sets of tags: $\mathbf{T}_t = \{T_{t,1}, T_{t,2}, ..., T_{t,n}\}$, where $n \leq m$. The whole process is summarised in Algorithm 1.

Algorithm 1: Tag meaning disambiguation

Input: Adjacency matrix $\mathbf{C}$ of the network of documents
Output: A set $\mathbf{T}$ of sets of tags

```
1  begin
2      // Document clustering;
3      X ← FastGreedyCommunityDiscovery(C);
4      T ← {};
5      // Extract top 10 tags;
6      for X_i ∈ X do
7          T_i ← Top10Tags(X_i);
8          T ← T ∪ {T_i};
9      end
10     // Merge similar sets of tags;
11     merged ← 1;
12     while merged = 1 do
13         merged ← 0;
14         for T_i, T_j ∈ T and i ≠ j do
15             if overlap(T_i, T_j) ≥ α then
16                 X_new ← X_i ∪ X_j;
17                 T_new ← Top10Tags(X_new);
18                 T ← T − {T_i, T_j};
19                 T ← T ∪ {T_new};
20                 merged ← 1;
21             end
22         end
23     end
24     return T;
25 end
```

4 Web Search Disambiguation

The result of the tag meaning disambiguation obtained from the method described in the previous section can be used to disambiguate Web search results. This is done by comparing the tags corresponding to the different meanings of an ambiguous tag with the keywords characterising a document in the search results. The steps are described in detail as follows.

Given a set D_t of documents returned by a search engine when queried with an ambiguous keyword t, our target is to classify the documents into different categories, each corresponding to a different meaning of t, yielding a set of sets of documents $\mathbf{D}_t = \{D_{t,0}, D_{t,2}, ..., D_{t,n}\}$ where $D_{t,0} \cup D_{t,1} \cup \cdots \cup D_{t,n} = D_t$. We assume that each document is only related to one meaning of t. Each set of documents $D_{t,i}$ corresponds to the meaning represented by T_i, except that $D_{t,0}$ is the set of documents which cannot be classified into any of these categories represented by $T_{t,1}, ..., T_{t,n}$. We further assume that each document $d_j \in D_t$ is characterised by a set $K_{t,j}$ of keywords, which could be the keywords used to

Algorithm 2: Web search disambiguation

Input: A set $\mathbf{T}$ of sets of tags, a set D_s of documents
Output: A set $\mathbf{D}$ of sets of classified documents

```
1  begin
2      // Initialisation;
3      D ← {};
4      for i ← 0 to |T| do
5          D_i ← {};
6          D ← D ∪ {D_i};
7      end
8      // Classify documents;
9      for d ∈ D_t do
10         x ← Cat_A(d);
11         D_x ← D_x ∪ {d};
12     end
13     return D;
14 end
```

index the document by the search engine, or the tags assigned to the document by users in a collaborative tagging system.

Firstly, we define the function *match* which calculates the extent to which the set $K_{t,j}$ of keywords of document d_j matches the set $T_{t,i}$ of tags of a particular meaning of the term t.

$$match(K_{t,j}, T_{t,i}) = \frac{|K_{t,j} \cap T_{t,j}|}{|T_{t,i}|} \tag{2}$$

By comparing the different values returned by the *match* function when different sets of tags are used, a document d_j is assigned to a particular category as follows.

$$Cat_A(d_j, t) = \begin{cases} \underset{i}{\text{argmax}}\ match(K_{t,j}, T_{t,i}), & \text{if } \max\limits_i match(K_{t,j}, T_{t,i}) \geq \beta \\ 0, & \text{if } \max\limits_i match(K_{t,j}, T_{t,i}) < \beta \end{cases} \tag{3}$$

The function Cat_A (the subscript A stands for automatic) assigns d_j a category which corresponds to the meaning of t represented by the set $T_{t,i}$ of tags which match the best with the keywords of d_j. However, if the keywords of d_j match poorly with any of the sets of tags, the document is assigned the category of 0. The threshold β is a value in the range of 0 to 1. The whole process is summarised in Algorithm 2.

5 Evaluation

In order to evaluate our proposed method of Web search disambiguation, we apply the method to Web search results obtained by querying del.icio.us and

Tag	Number of Documents	Number of Users
sf	426	446
tube	476	427
bridge	915	338
wine	421	896

Table 2. Statistics of the dataset collected from del.icio.us.

Google UK using four ambiguous terms, namely *sf*, *tube*, *bridge* and *wine*. These four terms are selected because it is observed that they are used to represent multiple concepts in del.icio.us, and that search results returned by Google when using these terms in the query also consist of documents related to a rather diverse topic. By applying the method on documents returned by del.icio.us, we can test whether our tag meaning disambiguation is able to identify all of the meanings of the ambiguous tags used in the system. On the other hand, by applying the method on documents returned by Google, we are able to study its performance on a traditional search engine.

5.1 Data Preparation

To generate the sets of tags representing the different meanings of the ambiguous terms, we collect data involving the four tags from del.icio.us by using a crawler program. The dataset includes documents which have been assigned the tags and users who have used the tags on the documents. In the data collection process, we skip documents which are only tagged by one user. Table 2 summarises the statistics of the dataset.

For Google UK, we submit queries using each of the four terms and obtain the top 50 pages returned. We denote the set of documents retrieved for the term t by GD_t. Del.icio.us, although primarily a collaborative tagging system, also provides search service on its data. However, search results returned by del.icio.us are ranked by how recent an item is tagged by a user instead of how relevant an item is to the keyword in the query. Hence, for each of the terms t we extract the top 50 items which are tagged by the greatest number of users with the tag in question as the search result and denote it by DD_t.

Finally, we construct a set of keywords for each document which are used to characterise the document. For documents returned by del.icio.us, the aggregated set of tags contributed by the users are used to form the set of keywords. For documents returned by Google, we first process the texts in the documents and extract keywords by filtering out stop words and non-text symbols, and then enrich the set by querying del.icio.us for the tags, if any, which are assigned to the documents.

Tag	Context	Tags Extracted
sf	San Francisco	sf, sanfrancisco, bayarea, san, francisco, california, travel, events, art, san_francisco
	Science fiction	sf, scifi, fiction, books, sci-fi, literature, writing, sciencefiction, science, fantasy
tube	YouTube videos	tube, youtube, video, funny, videos, fun, cool, music, feel.good, flash
	Vacuum tubes	tube, audio, electronics, diy, amplifier, amp, tubes, music, elect, guitar
	London underground	tube, london, underground, travel, transport, maps, map, uk, subway, reference
bridge	Design pattern	bridge, programming, development, library, code, ruby, tools, software, adobe, dev
	Card game	bridge, games, cards, game, imported, howto, conventions, card, bidding, online
	Computer networking	bridge, networking, linux, network, howto, software, sysadmin, firewall, virtualization, security
	Architecture	bridge, bridges, structures, engineering, science, physics, school, education, building, reference
wine	Software application	wine, linux, ubuntu, howto, windows, software, tutorial, emulation, reference, games
	Beverage	wine, food, shopping, drink, reference, vino, cooking, alcohol, blog, news

Table 3. Meanings of tags discovered and related tags extracted for each meaning.

5.2 Experiments

We first attempt to discover the different contexts in which the ambiguous tags are used by applying our proposed tag disambiguation algorithm on the del.icio.us dataset with $\alpha = 0.2$. By setting $\alpha = 0.2$, we effectively require two sets of tags to have more than three tags in common before we will combine them. This is based on the observation that very often the first three or four most frequently used tags in a set are sufficient for one to decide the meaning to which it corresponds.

The tags extracted for each of the ambiguous tags are shown in Table 3. We can see that the proposed algorithm performs well in revealing the multiple meanings of the tags. For example, four different meanings of the tag *bridge* are discovered, in which the tags extracted are closely related to the contexts in which *bridge* is used.

Next, we apply our proposed Web search disambiguation method, with $\beta = 0.3$, to the search results obtained from del.icio.us and Google. β is chosen based on a similar reason of the choice of α. We first manually classify the returned documents into the categories discovered in the tag meaning disambiguation phrase by inspecting their content. Our classification can be represented by a mapping $Cat_M(d, t)$ which assigns each document d a category x, where $x \in$

Tag	Case	Total	Classified	Unclassified	Classifiable	Correct	Precision	Recall	Coverage
sf	D	50	50	50	50	50	1.00	1.00	1.00
	G	50	38	12	38	37	0.97	0.97	0.74
tube	D	50	50	50	50	50	1.00	1.00	1.00
	G	50	34	16	33	31	0.91	0.94	0.62
bridge	D	50	43	7	49	42	0.98	0.86	0.86
	G	50	16	34	24	13	0.81	0.54	0.26
wine	D	50	50	50	50	50	1.00	1.00	1.00
	G	50	27	23	50	27	1.00	0.54	0.54

Table 4. Results of web search disambiguation. D stands for an experiment on del.icio.us-returned pages, while G stands for one on Google-returned pages.

$\{0, 1, 2, ..., n\}$. Category x corresponds to the meaning of the term t represented by the set $T_{t,x}$ of tags, and the category 0 is reserved for unclassified documents.

We evaluate the performance of the method by using three different measures, namely *precision*, *recall* and *coverage*. ***Precision*** measures the extent to which the documents are classified correctly. It is calculated by dividing the number of correctly classified documents by the total number of classified documents. ***Recall*** measures the fraction of classifiable documents which the method is able to classify. By classifiable documents we refer to documents which should fall into any one of the contexts discovered in the tag meaning disambiguation phase. Finally, ***coverage*** measures how many documents can be classified given the total number of documents returned. Let R_t be the set of retrieved documents, where $R_t = DD_t$ or $R_t = GD_t$ depending on the dataset on which we apply our algorithm. The three measures are defined as follows.

$$\text{Precision} = \frac{|\{d \in R_t | Cat_M(d,t) = Cat_A(d,t) \wedge Cat_M(d,t) \neq 0\}|}{|\{d \in R_t | Cat_A(d,t) \neq 0\}|} \quad (4)$$

$$\text{Recall} = \frac{|\{d \in R_t | Cat_M(d,t) = Cat_A(d,t) \wedge Cat_M(d,t) \neq 0\}|}{|\{d \in R_t | Cat_M(d,t) \neq 0\}|} \quad (5)$$

$$\text{Coverage} = \frac{|\{d \in R_t | Cat_M(d,t) = Cat_A(d,t) \wedge Cat_M(d,t) \neq 0\}|}{|R_t|} \quad (6)$$

The experimental results are shown in Table 4 and Figure 1.

5.3 Discussions

The experimental result shows that documents are classified to the correct categories the majority of the time, with precision ranging from 81% to 100%. This suggests that the tags extracted in the tag meaning disambiguation phase can be used to identify precisely the different contexts in which the ambiguous terms are used. Precision in the cases of del.icio.us was always higher than or equal to those in the cases of Google, probably because the documents in del.icio.us

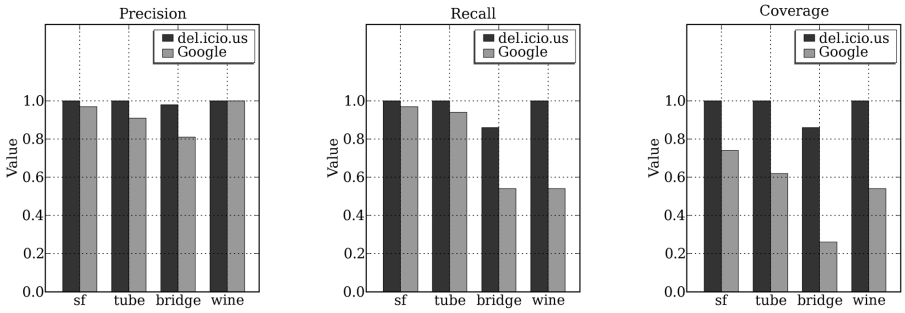

Fig. 1. Precision, recall and coverage of web search disambiguation.

feature keywords which are more similar to the tags used for disambiguation. After all, they are all contributed by users of del.icio.us.

Applying our technique to search results from del.icio.us generally results in higher recall (86% to 100%) than when we apply it to those from Google (54% to 97%). Low recall means that the algorithm is unable to classify documents which are actually related to one of the meanings discovered in the tag meaning disambiguation phase, probably due to poor matches between the tags extracted from del.icio.us and the keywords characterising the documents. This suggests that the tags extracted from del.icio.us may not be comprehensive enough to reconstruct the contexts in which those ambiguous words are used. Recall is particularly low when classifying documents returned by Google for *bridge* and *wine*. We find that quite a number of pages about bridges as architectural structures cannot be identified. These pages are characterised by keywords like *river*, *stream* and *architecture*, which are not present in the set of tags extracted from del.icio.us. Similarly, some pages about wine as a kind of beverage are not identified because they contain keywords like *red*, *white* and *bottle* which are absent from the set of tags for disambiguation. This problem is much less serious when classifying documents from del.icio.us, because the tags extracted are also the tags used frequently on these pages.

Performance in terms of coverage of our method on documents returned by del.icio.us is very satisfactory, suggesting that the tag meaning disambiguation method is able to identify all or most of the multiple meanings of the ambiguous tags used in del.icio.us. However, relatively low coverage (26% to 74%) can be observed in all the cases of classifying documents returned by Google. While low coverage is partly predicted by the low recall in these cases, this result also suggest that the tag meaning disambiguation process is not able to return the different meanings of an ambiguous tag used in a more general situation. For example, the common usage of *tube* to refer to a hollow and circular structure is not identified, which makes the Web search disambiguation algorithm unable to identify documents related to this meaning. On the other hand, among the documents returned by Google, there are in fact a certain number of documents

which are not related to any commonly known meanings of the query terms. For example, the coverage in the case of *bridge* is particularly low because some of the documents are only about places or organisations which are named *Bridge*. From this observation, we believe that a low coverage is not as undesirable as it seems, because the algorithm actually helps to filter out documents which are not semantically related to the query term.

In summary, our proposed method for Web search disambiguation is able to classify documents with high precision based on the implicit semantics extracted from collaborative tagging, though in some cases it is not able to identify all relevant documents for the categories. A major issue which requires further investigation is how to increase the comprehensiveness of the tags extracted from folksonomies in order to increase recall and coverage.

6 Related Work

To the best of our knowledge, this is the first study of the use of user-contributed annotations in collaborative tagging systems to disambiguate Web search results. Different methods have been used to discriminate word meanings or senses in the literature. These include the use of manual-constructed rules [9] and the use of dictionaries or thesauri [11, 12]. Our work is similar in part to studies which employ lexical co-occurrence to discover the different senses of an ambiguous word. For example, Schütze and Pedersen [17] derive a term vector for each word which represents word similarity derived from lexical co-occurrence. The vectors are then combined to form context vectors which are clustered to represent different senses of ambiguous words.

In addition, our work is also similar in principle to studies which apply document clustering techniques on Web search results. This is a problem quite extensively studied in the literature [3, 5, 19, 23] and is also addressed by commercial systems such as Vivisimo [10].[3] Existing document clustering techniques in general extract keywords from documents and calculate their similarity based on the keywords to obtain a set of clusters. Our approach differs from these techniques in that instead of performing clustering based on the vocabulary found in the documents returned by the search engine, we obtain a set of categories from analysis of collaborative tagging systems to aid classification of the documents. We believe our proposed method is better than existing approaches, as it is more focused in terms of the meanings of the keywords, while existing document clustering techniques might result in clusters which are not necessarily meaningful to the users.

On the other hand, while there have been no studies which directly address the problem of tag ambiguity, tag meaning disambiguation can be observed as a by-product in some research work which focuses on tag clustering. For example, in the work of Wu et al. [21] latent semantic analysis is applied to study the co-occurrence of tags, and ambiguous tags are found to score highly in multiple pre-

[3] The public version of Vivismo's Web search engine, Clusty, can be found at http://clusty.com/.

defined dimensions. Zhou et al. [24] also report that, in building a tag hierarchy by using deterministic annealing to perform tag clustering, tags with multiple meanings are found to appear in different branches of the resulting hierarchy. In addition, collaborative tagging is also used to improve Web search in general, such as by providing a better ranking of the search results [6, 22]. In contrast to these prior studies, our work directly addresses the problem of tag ambiguity, proposes a feasible solution and studies how the extracted semantics of tags can be applied to novel applications.

7 Conclusions and Future Work

In this paper, we propose a method for automatic Web search disambiguation which uses the implicit semantics extracted from folksonomies. Our preliminary evaluation shows that the tags extracted from tag meaning disambiguation can be used to classify search results returned by Web search engines with high precision. This suggests that tags contributed by users in collaborative tagging systems can be used to enhance the performance of Web search engines. Also, we note a distinct advantage of using tags extracted from collaborative tagging systems for Web search disambiguation. Our proposed method for tag meaning disambiguation is able to discover some unconventional meanings of the ambiguous words, such as *tube* for the video-sharing site YouTube, or *bridge* for the design pattern used in programming. These meanings are rather new and are of specific domains that they may not be available in dictionaries or thesauruses such as Wordnet [14], which are commonly used for word sense disambiguation in the literature [7].

At the same time, we are aware of several problems in the proposed method. In particular, the levels of recall and coverage are significantly lower than that of precision, meaning that some relevant documents cannot be identified with the tags we extract from a folksonomy. Based on the results reported in this paper, we plan to extend our research work in several directions. Firstly, we will study how the comprehensiveness of the set of tags which represents a particular meaning of an ambiguous term can be increased, such as by expanding it with tags which co-occur frequently with the set in order to increasing the chance of matching the keywords which characterise the documents. Secondly, we will investigate how we can identify more meanings of an ambiguous word to increase recall and coverage in Web search disambiguation, such as by complementing the contexts discovered in tag meaning disambiguation by information obtained from dictionaries. In addition, as our method for tag meaning disambiguation requires a post-processing step of combining clusters corresponding to the same meaning of a tag, we will also investigate how this process can be incorporated into the clustering process such as by considering other clustering algorithms. Finally, we will perform further evaluations which involve larger dataset and more ambiguous tags, in order to understand the performance of our proposed method in more general cases.

References

1. C. M. Au Yeung, N. Gibbins, and N. Shadbolt. Understanding the semantics of ambiguous tags in folksonomies. In Peter Haase, Andreas Hotho, Luke Chen, Ernie Ong, and Philippe Cudre Mauroux, editors, *Proceedings of the International Workshop on Emergent Semantics and Ontology Evolution (ESOE2007) at ISWC/ASWC2007, Busan, South Korea*, pages 108–120, November 2007.
2. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web (WWW7)*, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.
3. Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '92)*, pages 318–329, New York, NY, USA, 1992. ACM.
4. A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web (WWW2005)*, pages 902–903, New York, NY, USA, 2005. ACM.
5. "Peter Hannappel, Reinhold Klapsing, and Gustaf Neumann. MSEEC - a multi search engine with multiple clustering. In *Proceedings of the '99 Information Resources Management Association International Conference*, Hershey, Pennsylvania, May 1999.
6. Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *Proceedings of the 3rd European Semantic Web Conference (ESWC 2006)*, pages 411–426, 2006.
7. N. Ide and J. Véronis. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):1–40, 1998.
8. iProspect. Search engine user behaviour study, April 2006.
9. E. Kelly and P. Stone. *Computer Recognition of English Word Senses*. North-Holland, Amsterdam, The Netherlands, 1975.
10. Sherry Koshman, Amanda Spink, and Bernard J. Jansen. Web searching on the vivisimo search engine. *Journal of the American Society for Information Science and Technology*, 57(14):1875–1887, 2006.
11. R. Krovetz and W. B. Croft. Word sense disambiguation using machine-readable dictionaries. *SIGIR Forum*, 23(SI):127–136, 1989.
12. Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation (SIGDOC '86)*, pages 24–26, New York, NY, USA, 1986. ACM.
13. Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Journal of Web Semantics*, 5(1):5–15, 2007.
14. George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
15. M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.
16. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.
17. H. Schütze and J. Pedersen. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1995.

18. Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.
19. Jerzy Stefanowski and Dawid Weiss. Carrot^2 and language properties in web search results clustering. In *Proceedings of the International Atlantic Web Intelligence Conference (AWIC 2003)*, pages 240–249, 2003.
20. Thomas Vander Wal. Folksonomy definition and wikipedia. *http://www.vanderwal.net/random/entrysel.php?blog=1750*, November 2, 2005. Accessed 13 Feb 2008.
21. Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 417–426, New York, NY, USA, 2006. ACM Press.
22. Yusuke Yanbe, Adam Jatowt, Satoshi Nakamura, and Katsumi Tanaka. Can social bookmarking enhance search in the web? In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pages 107–116, New York, NY, USA, 2007. ACM.
23. Oren Zamir and Oren Etzioni. Grouper: a dynamic clustering interface to web search results. In *Proceeding of the eighth international conference on World Wide Web (WWW99)*, pages 1361–1374, New York, NY, USA, 1999. Elsevier North-Holland, Inc.
24. Mianwei Zhou, Shenghua Bao, Xian Wu, and Yong Yu. An unsupervised model for exploring hierarchical semantics from social annotations. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 680–693. Springer, 2007.

Introducing Triple Play for Improved Resource Retrieval in Collaborative Tagging Systems

Rabeeh Ayaz Abbasi and Ste en Staab

Department of Computer Science, University of Koblenz-Landau,
Koblenz, Germany
{abbasi,staab}@uni-koblenz.de
http://isweb.uni-koblenz.de/

Abstract. Collaborative tagging systems (like Flickr, del.icio.us, citeulike, etc.) are becoming more popular with passage of time. Users share their resources on tagging systems, and add keywords (called tags) to these resources. Users can search resources using these tags. But as the user gives more tags for search, he might not get su cient search results, because the resources might not be tagged with all the related tags.
We introduce the method *Triple Play*, which smoothes the tag space by user space for improved retrieval of resources. As a part of *Triple Play*, we also propose two new vector space models for collaborative tagging systems, *SmoothVSM Dense* and *SmoothVSM Sparse*. These vector space models exploit the user-tag co-occurrence relationship to overcome the problem of missing information in tagging systems. Finally we apply Latent Semantic Analysis to di erent vector space models and analyze the results. Initial experimentation show that using additional information available in tagging systems helps in improving search in tagging systems.

1 Introduction

Collaborative tagging systems provide their users an easy mechanism to store resources (like photos, bookmarks, publications) and add tags (keywords) to these resources. For example, a user can upload his photo of a trip to the beach of "St. Petersburg, Russia" to Flickr and tag it with *petersburg* and *beach*. He can search for the tags *petersburg* and *beach* to see this photo and other photos which are tagged with same tags by him or other users. Although tags provide an easy way to search resources, but they are only sparsely available. Many resources might not have all the relevant tags, and do not appear in relevant searches. If a user searches using less number of tags, he might get many undesired results, and if he provides many tags, he might a get few or no search results. Table 1 shows the number of search results for di erent tags searched on Flickr[1] website. Queries in table 1 assume boolean *AND* operator between the tags. It is obvious from Table 1 that as the number of tags in query increase, number of

[1] http://www.flickr.com/search/?m=tags

search results decrease rapidly. Consider a scenario in which a user has a lot of photos from Petersburg and Russia which have the tags *petersburg* and *russia*. Now he uploads some more photos of sunset at the beach of Petersburg, Russia. But he only adds the tags *beach*, *sunset*, and *sea* to these pictures. Now if someone searches these photos using tags *petersburg* and *beach*, he will not be able to retrieve these photos, because they do not have the tag *petersburg*. In this scenario, exploiting information about the tags which user has used would help in improving search results. This would not be possible by only searching the resources and their tags without considering user-tag information.

Table 1. Number of search results for tag queries searched at Flickr on February 15, 2008.

Tags Searched	Number of Results
petersburg	43,867
petersburg, beach	797
petersburg, beach, russia	7
petersburg, beach, russia, sea	4
petersburg, beach, russia, sea, sunset	0

Currently, collaborative tagging systems provide tag search based on simple tag matching. The search results might not be very satisfying due to the problem of sparsity in the data (i.e., less amount of information available in tagging systems). Most of the information retrieval approaches inherently work on two dimensions, that are documents (resources) and terms. But in case of collaborative tagging systems there are also other dimensions like user information.

We introduce *Triple Play* to improve search results. *Triple Play* overcomes the sparsity of information by using further information available in tagging systems. Speci cally, it uses tag-resource and user-tag relationship information. Using user-tag relationship information helps in smoothing the information which is otherwise not available in simple tag-resource relationship. In *Triple Play*, we propose two vector space models *SmoothVSM Dense* which uses user-tag covariance information and *SmoothVSM Sparse* which considers users as resources.

Once we have an appropriate VSM for tagging system, we use Latent Semantic Analysis (LSA) [3] to provide better search in tagging systems. LSA reduces dimensions of a vector space which helps in overcoming the problem of sparsity in the data. Initial experimental results show that using additional information available in tagging systems and LSA helps in improving search in collaborative tagging systems. Figure 1 shows the overall process of *Triple Play*.

In next section we formally describe collaborative tagging systems, our proposed vector space models, and how we use them to improve search in collaborative tagging systems.

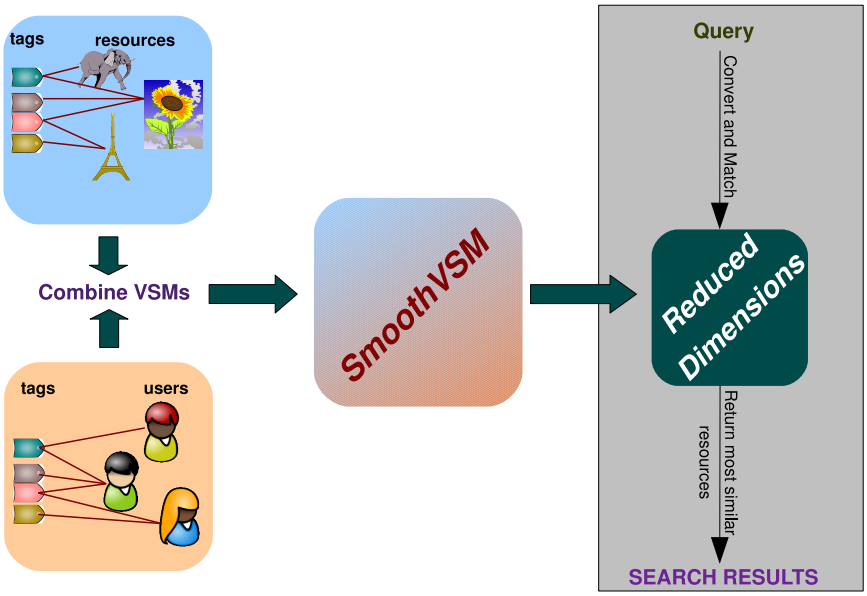

Fig. 1. Overall process of *Triple Play*.

2 Method

We start with the formal representation of Collaborative Tagging Systems

2.1 Formal Representation of Collaborative Tagging Systems

We use the same formal de nition for de ning tagging systems as a tripartite graph between users, tags, and resources given by [5]. Let us de ne the collaborative tagging system S of users, tags, and resources, and relationship between users, tags, and resources as a quadruple

$$S = (U, T, R, Y) \tag{1}$$

where U represents set of users, T represents set of tags, R represents set of resources and $Y \subseteq U \times T \times R$ is ternary relation over U, T and R. If a user $u \in U$ uses the tag $t \in T$ to tag a resource $r \in R$, then there is a relation $(u, t, r) \in Y$.

2.2 Tag Frequency Normalization

In standard information retrieval (IR) tasks, normalization techniques like Term Frequency Normalization are used. Term frequency normalization is used to prevent bias towards longer documents. We use the same idea of term frequency

normalization in tagging systems for resources and users. Tag frequency normalization for resources prevents bias of results towards resources having a large number of tags.

Let us de ne the number of times a tag t appears with a resource r as frequency of the tag t with resource r. We represent tag frequency based on resources with the function $f_r(t)$ which returns number of times a tag t appears with a resource r.

$$f_r(t) = |\{(u,t,r) \in Y, u \in U\}| \tag{2}$$

In some tagging systems (called *Narrow Folksonomies* [7] like Flickr), a resource cannot be tagged with a tag more than once, while in other tagging systems (called *Broad Folksonomies* [7]) a single resource can be tagged with a tag multiple times (for example from di erent users). In case of *Narrow Folksonomies*, the function $f_r(t)$ will always return the value 1 or 0.

We normalize the frequencies of tags by dividing occurrences of a tag in a resource by total number of tag occurrences of that resource. Normalized tag frequency $tf_r(t)$ of a tag t in a resource r is de ned as follows

$$tf_r(t) = \frac{f_r(t)}{\sum f_r(t')}, (u,t,r) \in Y, (u,t',r) \in Y \\ , t' \in T, u \in U \tag{3}$$

To improve search results, we want to use the user-tag information present in collaborative tagging systems. For this reason, we need to compute tag frequencies based on user-tag relationship. We de ne frequency of tag based on user, the function $f_u(t)$ gives the number of times user u has used the tag t.

$$f_u(t) = |\{(u,t,r) \in Y, r \in R\}| \tag{4}$$

As we normalize the tag frequencies based on resources, we also normalize the tag frequencies based on users. This normalization reduces the bias of search results towards users who use a large number of tags. Normalized tag frequency $tf_u(t)$ of a tag t based on a user u is de ned as follows

$$tf_u(t) = \frac{f_u(t)}{\sum f_u(t')}, (u,t,r) \in Y, (u,t',r) \in Y \\ , t' \in T, r \in R \tag{5}$$

2.3 Vector Space Models (VSMs)

Now we de ne Vector Space Model (VSM) based on de nitions of previous sections. First we de ne a simple VSM based on tag-resource relationship, which is analogous to term-document matrix in traditional information retrieval. It is represented as a matrix X^f with $|T|$ rows and $|R|$ columns, where each row represents a tag vector and each column represents a resource vector. t, r element of the matrix X^f represents number of times tag t is used with resource r.

$$X^f(t,r) = f_r(t) \tag{6}$$

We also de ne normalized VSM, X based on tag-resource relationship as follows.

$$X(t,r) = tf_r(t) \tag{7}$$

Similarly we de ne user based VSM, W^f and user based normalized VSM model W as follows.

$$W^f(t,u) = f_u(t) \tag{8}$$

$$W(t,u) = tf_u(t) \tag{9}$$

where element at location t, u in the VSM, W^f represents number of times user u has used the t tag. W is the normalized form of the VSM, W^f.

Once we de ne VSMs and normalized VSMs based on tag-resource and user-tag relationships separately. We now de ne *SmoothVSMs* which are based on tag-resource and user-tag relationship simultaneously. To include user-tag information to VSMs de ned previously, rst we compute normalized co-variance of tags based on users by multiplying the normalized VSM based on user-tag relationship with its transpose $W * W'$. Our hypothesis for computing co-variance based on users is that, it will group tags based on users usage of tags. For example if a user has used tags *Petersburg* and *Sea*, and these two tags do not appear in any resource together. After multiplication, these two tags will have some co-occurrence value which might not be obvious otherwise and this will help to improve search. Now to create the *SmoothVSM Dense*, based on tags and resources (which will be used for searching resources), we multiply this user based tag co-variance matrix with normalized VSM of tag-resource X. As a result of all these multiplications, we get a *SmoothVSM Dense* which represents tag-resource relationship but also contains information of user-tag relationship. Despite of its name, *SmoothVSM Dense* is still a sparse VSM, but it is much denser as compared to other VSMs. *SmoothVSM Dense* based on normalized VSMs of tag-resource X and user-tag co-variance $W * W'$ is de ned as follows

$$Z = W * W' * X \tag{10}$$

We propose another VSM called *SmoothVSM Sparse* which also considers user-tag information. We consider the users in the tagging systems as resources. To create a VSM based on this assumption, we augment the normalized VSM based on user-tag relationship W to normalized VSM based on tag-resource relationship X. Such that rst $|R|$ columns of the new normalized augmented VSM Q represent resource vectors and last $|U|$ columns represent user vectors. We de ne *SmoothVSM Sparse* Q with $|T|$ rows and $|R| + |U|$ columns using matrix augmentation operator $|$ as follows

$$Q = (X|W) \tag{11}$$

After de ning standard VSM X, *SmoothVSM Dense* Z, and *SmoothVSM Sparse* Q. In next section we describe how can we apply Singular Value Decomposition (SVD) to these VSMs.

2.4 SVD for Improving Search

In Singular Value Decomposition (SVD), a matrix M is decomposed into three matrices[2] L, G, H.

$$M = L * G * H' \tag{12}$$

Where L and H are called left and right singular matrices respectively. Columns vectors of L and also H are orthogonal to each other, that means dot product of same vector in a matrix (L or H) results in 1 and dot product of two di erent vectors is always 0. Columns of matrices L and H are also called *eigen vectors*. Matrix G, called singular matrix, is a diagonal matrix with singular values at its diagonal in descending order.

We can approximate the original matrix M by multiplying rst k column vectors of matrix L, rst k singular values of matrix G and rst k rows of matrix H'. This is also equal to reducing the dimensions of original matrix. This reduction of dimensions helps in reducing noise present in original data. Approximation of original matrix is de ned as follows

$$M \approx M_k = L_k * G_k * H'_k \tag{13}$$

Latent Semantic Analysis (LSA) [3] reduces dimensions for better information retrieval. In case of collaborative tagging systems, the matrix M is one of the VSMs de ned in Section 2.3. M_k is the approximation of the original VSM. Rows of the matrix L_k represent tag vectors in k reduced dimensions, and columns of L_k represent *latent tags*. Similarly columns of the H'_k matrix represent resources and rows of H'_k represent *latent resources*. We can compute similarity between rows of L_k matrix to retrieve similar tags and columns of H'_k to retrieve similar resources. To retrieve resources against a query, we consider the query as a resource and convert the query into reduced dimensions. Then we compute its similarity with the resources (column vectors in H'_k matrix) to retrieve the most similar resources to the query.

In case of *SmoothVSM Sparse* (Q), we have more rows than the resources in the H'_k matrix. First $|R|$ columns of augmented H'_k matrix represent resources and last $|U|$ columns represent users. As we want to retrieve only resources against a query, therefore we only consider rst $|R|$ columns of H'_k. Our assumption is that the reduced dimensions of matrix H'_k includes information about tag-resource and user-tag relationships.

2.5 Querying and Retrieval

To retrieve resources from a VSM against a query, we have to represent the query as a vector (similar to a resource vector). Let q represent the query (a column vector of length $|T|$) with all of its elements equal to zero except for

[2] Standard notation for SVD uses the matrices U, S, and V instead of L, G, and H respectively, but because we use U and S for representing users and tagging system respectively, therefore we do not use standard symbols used for SVD

those elements at indexes which are indexes of the queried tags in VSM. To retrieve resources against a query without applying SVD, we compare q with the column vectors of VSM (X, Z, or Q) using cosine similarity (equation 15).

But to retrieve resources against a query in reduced dimensions (i.e., after applying SVD), we have to convert the query q in reduced dimensions. Query q is converted into reduced dimensions q_k as follows [3]

$$q_k = q' * L_k * {G_k}^{-1} \tag{14}$$

Now we compare the reduced query q_k to the column vectors of H'_k and retrieve the most most similar resources to the query. We compute the similarity between two vectors a and b using cosine similarity as follows

$$cosine(a, b) = \frac{a \cdot b}{\|a\| \cdot \|b\|} \tag{15}$$

If vectors a and b are same, then their cosine similarity is equal to 1 and if vectors a and b have no common term, then their cosine similarity is equal to zero.

In next section, we describe di erent experiments using VSMs de ned in Section 2.3.

3 Dataset and Evaluation

In this section, rst we describe the dataset we used for our experiments and then the evaluation method.

3.1 Data set

For experiments, we create a dataset of 10000 random resources uploaded to Flickr between 2004 and 2005. This dataset contains information about 8707 users, 10000 photos, 18435 tags, and 39775 taggings. We do not apply any kind of ltering on the dataset. The dataset contains many resources using more than 50 tags (e.g. a photo at Flickr website[3]) and also many resources using only one tag.

3.2 Evaluation Method

In ideal case, evaluation for our approach would be human based, because a human user can tell whether the results are related to the query or not. We plan to do human based evaluation of our approach. For initial experimentation we create the scenario of querying and retrieval arti cially. Our assumption are that, the resources do not have all the relevant tags, and as the size of the query exceeds, the query returns less number of resources. To test this hypothesis, we

[3] http://www. ickr.com/photo.gne?id=78192499

take the given dataset as gold standard and to derive a test dataset from the gold standard, we remove some tags from a resource and create VSM without removed tags in that resource. By removing tags from a resource, we know that the removed tags belong to this resource, but this information is not available in the VSM, our goal is to retrieve this resource. We make a query from removed tags of a resource and remaining tags of that resource. Then we count the number of resources n that match this query in the gold standard dataset. For good retrieval, the resource from which the tags were removed, shall be retrieved in rst n search results, because n resources have the queried tags. If the document is retrieved in rst n search results, we say that the query is matched. Otherwise we say that the query is not matched. We do this procedure with di erent number of removed tags and query lengths. The procedure of creating gold standard dataset is de ned as follows

1. Select r random resources for creating queries
2. Randomly remove m tags from each of the r selected resources, called missing tags. (Note that, these removed tags remain in the gold standard dataset)
3. For each of the r resources, create a query using m missing tags and p remaining tags
4. For each query $i, i = 1..r$, lets say there are n_i resources in the gold standard dataset

Once we have gold standard dataset, now we create the test dataset

1. Create VSM (using a method described in Section 2.3) without tag-resource information about m missing tags in each of the r resources
2. Retrieve resources for query i and sort them on the basis of similarity
3. If the resource from which query i was made, appears in rst n_i results of that query, then we consider it a matched query, otherwise an unmatched query

After calculating the number of matched queries, we compute precision as follows

$$Precision = \frac{Number\ of\ Matched\ Queries}{Total\ Number\ of\ Queries} \tag{16}$$

In next section we discuss the results of experiments we perform

4 Results and Discussion

For the selected dataset, we create gold standard and test dataset considering 1000 random queries using the method described in Section 3.2. Each query has m missing tags and one remaining tag. Length of the query is calculated by adding number of missing and remaining tags. We apply SVD on each VSM and the convert queries to reduced dimensions (using equation 14) before computing similarity and retrieving resources.

Figure 2 shows the precision of search results using di erent VSMs (described in Section 2.3) after applying SVD for di erent query lengths. Search results for

the *SmoothVSM Dense* (Z) are better than others for queries of all lengths. This is because of the reason that *SmoothVSM Dense* is enriched with a lot of information. Due the the multiplication process, *SmoothVSM Dense* (Z) also includes the information which would not be present in simple VSM based on tag-resource relations like X. The *SmoothVSM Sparse* (Q) performs better than simple VSM (X) because *SmoothVSM Sparse* still has the information which is missing in X, i.e., information about user-tag relationship. Although *SmoothVSM Sparse* and *SmoothVSM Dense* both contain information about user-tag relationship, but *SmoothVSM Dense* performs better due to the grouping of tags based on users. No such grouping is done in *SmoothVSM Sparse*. In *SmoothVSM Sparse* the users are just considered as additional resources.

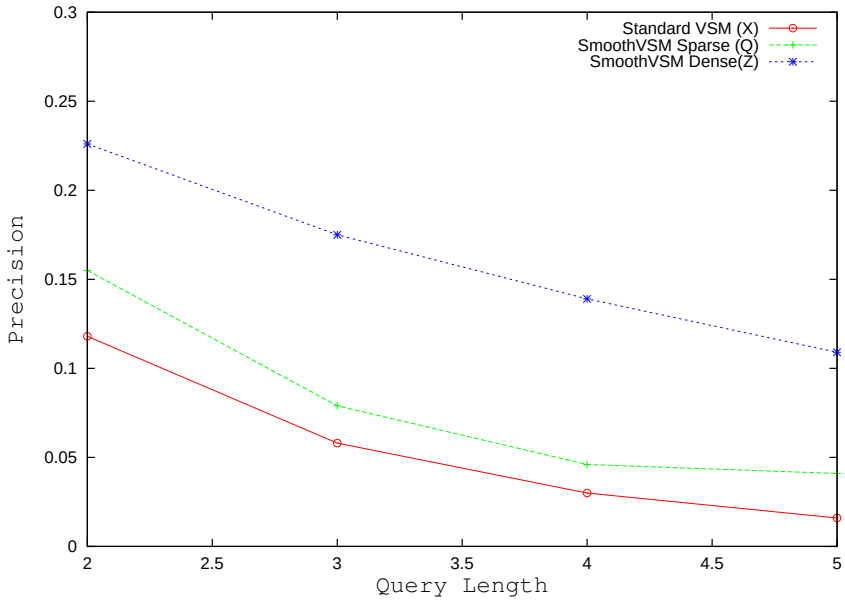

Fig. 2. Precision of search results with increasing query length. Results are displayed after applying SVD to all the three VSMs. 200 singular values were used for each VSM.

Figure 3 shows the e ect of number of singular values (number of dimensions) on precision. If we select very few singular values, then the precision of retrieved search results becomes low. Selecting a high number of singular values is also not a very good decision, because doing SVD for higher dimensions has more computational costs. For example, doing SVD of *SmoothVSM Sparse* (Q) for 800 singular values requires 3 hours of time on a 2.00 GHz processor, but for 20 singular values, it requires only 9 seconds on the same machine. For the given dataset, 200 to 400 singular values is a good compromise between quality and computational time.

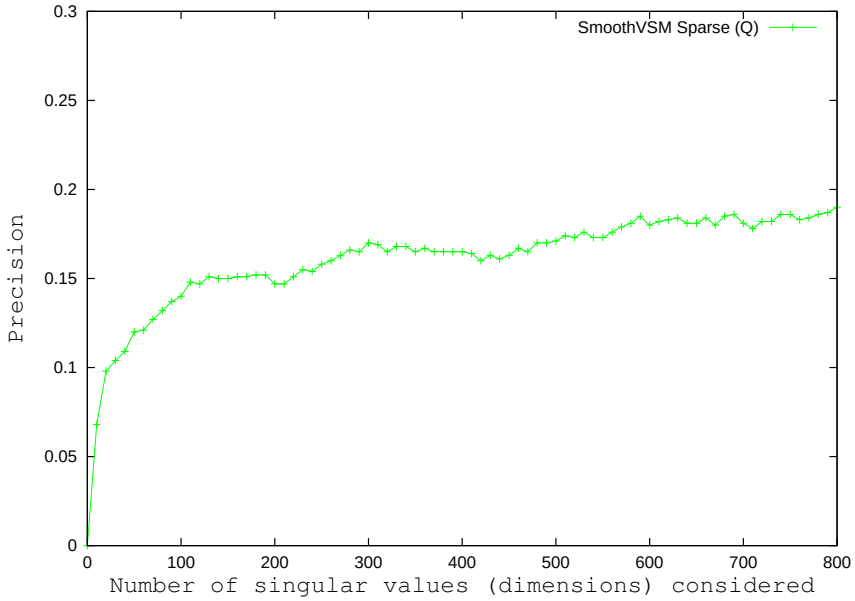

Fig. 3. Precision of search results by increasing number of singular values.

5 Related Work

Searching in collaborative tagging systems is becoming an interesting research area. [4] design a task to search particular resources on Flickr website for iCLEF 2006. They present di erent experiment results for the task. [5] present a search algorithm Folkrank for searching resources in tagging systems. They search resources based on popularity of tags. [1] cluster tags to improve the exploration experience of user. For a given tag, they nd other similar tags. They focus on improving search experience by exploring related tags. Our method di ers from these approaches in the way we enrich VSM and use Latent Semantic Analysis (LSA) to improve search results.

We use multiplication in one of the Vector Space Models (VSMs) to enrich the nal VSM with more information (including user-tag relationship information) to improve search in collaborative tagging systems. Similar kind of idea is used by [6] to create community based light weight ontologies. They call the covariance matrix obtained by $W * W'$ (see Eq. 10), a light weight ontology. Their focus is on extracting light weight ontologies from tagging systems.

LSA has been used for improving information retrieval tasks. [3] describe how Singular Value Decomposition (SVD) can be used for better information retrieval. [2] augments features to existing VSM to improve classi cation process. Their focus is to improve classi cation process using augmented features. [8] de ne a general framework of applying LSA on multiple co-occurrence relationships.

6 Conclusions and Future work

In this paper we show that how can we improve search in collaborative tagging systems like Flickr or del.icio.us, particularly when there are more tags in the search query. We formally de ne collaborative tagging system and propose a method *Triple Play*, in which we create a vector space model (*SmoothVSM Dense* or *SmoothVSM Sparse*) considering user-tag relationship information available in collaborative tagging systems and then apply Latent Semantic Analysis for retrieval of resources from these vector space models. We evaluate our proposed method by arti cially removing information from data and then searching for it. This approach is not the best method to evaluate, therefore we plan to do human judged evaluation of our methods. Initial experiments show that we can use *Triple Play* to improve search in collaborative tagging systems. We plan to do experiments by assigning di erent weights to vector space models and combining these vector space models.

7 Acknowledgments

This work has been partially supported by the European project *Semiotic Dynamics in Online Social Communities* (Tagora, FP6-2005-34721). We would like to acknowledge Higher Education Commission of Pakistan and German Academic Exchange Service (DAAD) for providing scholarship and support to Rabeeh Abbasi for conducting his PhD.

We thank Bhaskar Mehta, members of L3S group Hanover and Prof. Dr. Klaus Troitzsch for discussions related to this work and Klaas Dellschaft for providing Flickr data.

References

1. G. Begelman, P. Keller, and F. Smadja. Automated Tag Clustering: Improving search and exploration in the tag space. *Proc. of the Collaborative Web Tagging Workshop at WWW*, 6, 2006.
2. S. Chakraborti, R. Mukras, R. Lothian, N. Wiratunga, S. Watt, and D. Harper. Supervised Latent Semantic Indexing Using Adaptive Sprinkling. *Proceedings of IJCAI*, pages 1582–1587, 2007.
3. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
4. J. Gonzalo, J. Karlgren, and P. Clough. iCLEF 2006 Overview: Searching the Flickr WWW photo-sharing repository. *Proceedings of CLEF*, page 8, 2006.
5. A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. *Information Retrieval in Folksonomies: Search and Ranking*, pages 411–426. 2006.
6. P. Mika. Ontologies are us: A uni ed model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):5–15, 2007.
7. T. V. Wal. Explaining and showing broad and narrow folksonomies, 2005. Available at `http://www.personalinfocloud.com/2005/02/explaining_and_.html`.

8. X. Wang, J.-T. Sun, Z. Chen, and C. Zhai. Latent semantic analysis for multiple-type interrelated data objects. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 236–243, New York, NY, USA, 2006. ACM.

Keyword Suggestion Using Concept Graph Construction from Wikipedia Rich Documents

Hadi Amiri, Abolfazl AleAhmad, Masoud Rahgozar, Farhad Oroumchian
Database Research Group, School of Electrical and Computer Engineering
Campus #2, University of Tehran, North Kargar St., Tehran, Iran.
{h.amiri, a.aleahmad}@ece.ut.ac.ir, rahgozar@ut.ac.ir, oroumchian@acm.org

ABSTRACT
Concept graph is a graph in which nodes are concepts and the edges indicate the relationship between the concepts. Creation of concept graphs is a hot topic in the area of knowledge discovery. Natural Language Processing (NLP) based concept graph creation is one of the efficient but costly methods in the field of information extraction. Compared to NLP based methods, Statistical methods have two advantages, namely, they are language independent and more computationally efficient. In this paper we present an efficient statistical method for creating a concept graph from a large document collection. The documents which are used in this paper are from Wiklipedia collection because of their rich and valid content. Moreover, we use the final concept graph to suggest a list of similar keywords for each unique concept or combination of concepts to find deeper information to help information extraction. Also, we will show the viability of our approach by comparing its result to a similar system called the *Wordy* system.

Keywords
Concept Graph, Keyword Suggestion, Concept Relation.

1. INTRODUCTION
Knowledge representation is an issue that is relevant to both cognitive science and artificial intelligence. In the area of cognitive science, knowledge representation is concerned with how people store and process information. In artificial intelligence (AI) the primary aim is finding efficient methods to store knowledge so that programs can process and manipulate it. AI researchers have borrowed representation theories from cognitive science. One such approach is concept graph or CG. A Concept Graph is a graph in which nodes are concepts and the edges indicate the relationship between the concepts [2].

In order to gain good results the construction of concept graphs should be done efficiently and effectively. NLP-based and statistical approaches are two approaches for this task. Statistical approaches are computationally more efficient than NLP-based approaches; However NLP-based approaches are effective. In this paper we present a statistical approach that has the advantage of being language independent and more computationally efficient. The richness of the source text has a significant impact on the quality of the Concept Graph representation of the text. Since Wikipedia has valid and very rich content, we have experimented with the Wikipedia collection in our tests.

1.1 Concept Graph
Concept graphs are not intended as a means of storing data but as a means of describing data and the interrelationships. As a method of formal description, they have three principal advantages: First, they can support a direct mapping onto a relational data base; second, they can be used as a semantic basis for natural language; and third, they can Support automatic inferences to compute relationships that are not explicitly mentioned [2]. The third point is the principal topic of this paper.

Concept graphs can be used for different purposes, for example Ardini et al. used them for query expansion [11] and Kang et al. used them for Web-Document filtering [12]. In this research in addition to construction of a concept graph we will discuss using the graph for keyword suggestion, namely, having a concept we can suggest the terms/keywords related to that concept. To have a sample result of the proposed system we will show some keywords suggested by our system and compare them with the keywords suggested by Wordy system [1]. Wordy is a framework for keyword generation for search engine advertising. This framework uses semantic similarity between terms to find the terms relationships.

The rest of the paper is organized as below. Section 2 describes the collection we have used in this research. Section 3 explains the steps we followed one by one to construct the concept graph as the outcome of this paper. In this section we precisely explain the tuning parameters and other algorithms used in this research. In Section 4 we compare our system with Wordy.

2. WIKIPEDIA COLLECTION
In this research, the INEX 2006 Wikipedia collection [8] is used. As we know the content of the Wikipedia documents is rich that fits our purpose. Furthermore, this collection is general and nearly up to date (2006) that help us to increase the generality of the results. Table 1 shows some statistical information about the INEX 2006 Wikipedia collection.

Table 1. INEX 2006 Wikipedia Collection Information

Feature	Value
Index Size	2.24 GB
Number of Docs	+658,000
Number of Terms	+267,625,000
Number of Unique Terms	+3,540,000

The number of nodes in the final concept graph is nearly the same as the number of unique terms in the collection. However some useless terms will be removed form the final graph. We will explain the removal process in the subsequent section. For stemming, stop-word removal, indexing and retrieval purpose we

used the lemur toolkit[1]. This open source search engine is one of the best toolkits designed to facilitate research in language modeling and information retrieval. Lemur supports indexing of large-scale text databases, the construction of simple language models for documents, queries, or sub-collections, and the implementation of retrieval systems based on language models as well as a variety of other retrieval models.

3. CONCEPT GRAPH CONSTRUCTION

In this research we want to create a concept graph using recursive vector creation method. This section explains the steps we followed to create this graph.

3.1 Clustering Method

The document clustering techniques are for partitioning a given data set into separate clusters, with each cluster composed of the documents with similar characteristics. Most existing clustering methods can be broadly classified into two categories: partitioning methods and hierarchical methods. Partitioning algorithms attempt to partition a data set into k clusters such that a previously given evaluation function can be optimized. The basic idea of hierarchical clustering methods is to first construct a hierarchy by decomposing the given data set, and then use agglomerative or divisive operations to form clusters. In general, an agglomeration-based hierarchical method starts with a disjoint set of clusters, placing each data object into an individual cluster, and then merges pairs of clusters until the number of clusters is reduced to a given number k. On the other hand, the division-based hierarchical method treats the whole data set as one cluster at the beginning, and divides it iteratively until the number of clusters is increased to k [4].

In this research we used the EM clustering algorithm that is also developed as a part of Weka open source toolkit [7]. Given a model of data generation and data with some missing values, EM uses the current model to estimate the missing values, and then uses the missing value estimates to improve the model. Using all the available data, EM will locally maximize the likelihood of the generative parameters giving estimates for the missing values. This algorithm is a partitioning algorithm and generates probabilistic descriptions of the clusters in terms of mean and standard deviation. This method is used widely for the data clustering purposes [4, 6].

3.2 Representative Vector Creation

3.2.1 Initial Terms Selection

This section explains the steps we followed to create representative vectors for each concept in the collection. We consider each term in Wikipedia collection as a concept. However we have a removal process that removes useless concepts.

Figure 1 shows the system architecture. The process starts with a query q. This query is a random single term from the Wikipedia collection. We consider each query to represent an initial concept and try to find other concepts related to this one from the collection.

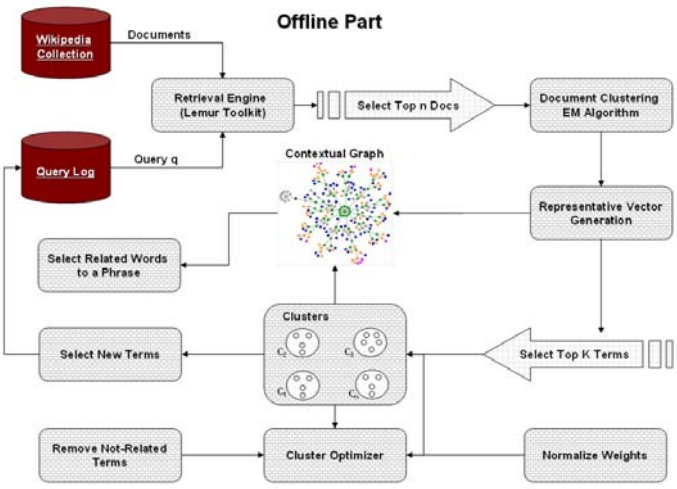

Figure 1. System Architecture

The initial retrieval step ranks the retrieved documents in decreasing order of query-document similarities and creates a ranked list for each query. Then we use EM clustering algorithm provided by Weka in order to detect different contexts of the retrieved documents and group them. As the authors in [6] suggest, there is not a statistically significant variation in query-specific cluster effectiveness for different values of top-ranked documents, hence we use top-10 documents for the context detection purpose. The result of this step is documents and their related context clusters for each query.

Next, the system should generate a terms vector to represent each cluster. The most popular frequency based term ranking methods are TF (term frequency) and TF/IDF (term frequency/inverted document frequency) [5]. The TF/IDF penalizes the weights for common keywords that appear in large number of documents. This measures works well on clustering text documents and we used this weighting schema to assign the degree of relationship between documents' terms and queries. This weighting scheme is shown in Equation (1).

$$w_{d,t_i} = \frac{tf(t_{i,d}) * (C_{doc} - df(t_i))}{\sum_i tf(t_{i,d}) * C_{doc}} \quad (1)$$

In above Equation $w_{d,ti}$ is the weight of term t_i in the document d. This weight shows the degree of relationship between documents' terms and query. $tf(t_i, d)$ is the frequency of the term t_i in the document d, $\sum tf(t_i, d)$ is the length of the document d, $df(t_i)$ is number of documents contains term t_i and C_{doc} is the total number of documents in the collection.

The *representative vector* is a vector that contains related terms/concepts and the degree of relationship between these terms and the query. Each query may have more than one representative vector, because the query may have different clusters (contexts) determined by the EM clustering algorithm. In order to create the initial representative vector, we normalize the weights of each term in a document. Equation (2) is used for this purpose.

[1] www.lemurproject.org/

$$w'_{d,t_i} = \frac{w_{d,t_i} - Min(w_{d,t})}{Max(w_{d,t}) - Min(w_{d,t})} + c \qquad (2)$$

In the above equation $Min(w_{d,t})$ and $Max(w_{d,t})$ are the minimum and maximum term weights in document d respectively and $w_{d,ti}$ is the weight of term t_i in the document d computed by TF/IDF scheme. After this normalization the weights would come into the range [0, 1]. The value of c is set to a small value to prevent zero weights and for all $w'_{d,ti}$ we set $w'_{d,ti}$ to one. This normalization makes the weights of the terms in different documents to be comparable to each other. In the next step we create a pool of all the terms in each cluster to select the most important representative terms for the cluster. Before the selection, we re-normalize all the weights of the terms in the pool according to Equation (3):

$$w_{t_i} = \frac{\sum_{j=1}^{NoDocs} w'_{d_j,t_i}}{NoDocs} \qquad (3)$$

Where $w'_{dj,ti}$ is the weight of the term t_i in document d_j and *NoDocs* indicates the number of documents in the cluster. If a document doesn't have the term t_i, we consider the weight of the term t_i to be zero in that document. This normalization increases the weights of the terms that appear in more documents and decreases it for the less frequent terms. Then, we choose top 100 terms with highest weights in the pool as an *initial representative vector* for each cluster. Till now, for each query we cluster the retrieved documents for that query and create a representative vector for each cluster, so each query could have several clusters with their representative vectors. However the cluster optimizer part decreases the number of these vectors.

3.2.2 Representative Vector Optimization

This section describes a part of the architecture that is used to optimize the representative vectors. As it is shown in Figure 1, the proposed architecture contains two parts to optimize the representative vectors of the clusters. To make the vector stronger, we define the following principle:

Principle 1: *If there was a relation between two terms, this relation is association relation and should be bidirectional.*

This means, if concept 'a' exists in the representative vector of concept q, then the concept 'q' should appear in the representative vector of the query a. On the other hand, if the relation between a query, each query is expressed by a single term, and its related term were a unidirectional relation, this means the relation is not strong enough and the term will be removed from the representative vector of the query. Constructing the cluster using the above principal improves the representative vectors quality by selecting highly related terms. This method removes some terms from the vectors; we named these terms *not-related terms*. Hence, the weights of terms in the vectors should be renormalized.

Let us formalize the entities involved in this activity. We indicate by q a concept expressed by a single term. Also, let $T=\{t_0,t_1, \ldots ,t_{100}\}$ and $W_T=\{w_0,w_1, \ldots, w_{100}\}$ be the *initial representative vectors* of the query q. In other words, the T vector contains related terms to query q in one of its clusters and the vector W_T contains its corresponding weights. Imagine term t is a *not-related term* to q appeared in the vector T. To automatically detect this term we first create an initial representative vector for each term in q's representative vector, the same process as the system did for query q. This means we do a search again with each term in q's representative vector as a separate query and then cluster the output of each and then build representative clusters for each query term. Then we follow *Principal 1* to find not-related terms in q's representative vector. Because the term t is a not-related term to query q, the representative vector of this term, will not contain q. Hence, the term t will be removed from vector T. However if the relation between t and q were a bidirectional relation, we should follow Equation (4) to choose a new weight for the relation of terms and update weight vectors:

$$w_t = \frac{Max(w_{t,q} + w_{q,t})}{2} \qquad (4)$$

In which $w_{q,t}$ is the weight of the relation from q and t (when t is in the representative vector of q) while $w_{q,t}$ is the weight of the relation from t to q (when t is in the representative vector of q). The Max operation is used because the terms may appear in more than one representative vector of queries.

After removing *not-related terms* from the vectors and finding new weights, we should renormalize the weights in the representative vectors. To do so, we apply the following equations one by one:

$$w'_{t_i} = \frac{w_{t_i} - Min(w_t)}{Max(w_t) - Min(w_t)} + c$$

$$w''_{t_i} = AVG(w_t) - w'_{t_i} \qquad (4)$$

$$w'''_{t_i} = Max(w'_{t_i}, w''_{t_i})$$

In the above equation $Min(w_t)$ and $Max(w_t)$ are the minimum and maximum term weights in the representative vector, W_T. Using this normalization the weights will become in range [0, 1]. Again the value of c is set to a small value to prevent w'_{ti} when $w'_{ti}=Min(w_t)$ and for all $w'_{ti}>1$ we set $w'_{ti}=1$. Using the second equation, we adjust the weights of the low weight terms to give them the chance to contribute in the related cluster especially if they appear in most of the retrieved documents. This weighting is a kind of fuzzy weighting schema [13, 9].

It should be mentioned that the *representative vectors* for each concept are created once. Having these vectors we are able to create the final concept graph. We use this graph to find deeper information of concepts to help information extraction. In this research we use the graph for the purpose of word suggestion; namely, we suggest a subset of the graph concepts that are more similar to the query's terms.

4. EVALUATION

This section presents the result of the proposed system for some sample terms. In our experiments, we consider queries used in [1] and compare the results of our system with the results of that system that is named *Wordy* [1]. They used five queries to study

the result of *Wordy*: *Skin*, *Teeth*, *Pedicure*, *Massage* and *Medical*. Among them, the third query term is a special scientific term and has no relevant document in the Wikipedia collection, so we could not consider that query in our experiments. Table 3 at Appendix 1 shows keywords suggested by *Wordy* and Our system for the four queries. In [1] the authors, for the sake of brevity, only listed the top 10 suggestions generated by *Wordy,* and we do the same here to prepare a comparable view of results. The description column describes the words retrieved by our system to show how selected words are related to the query.

As it can be understood from Table 3, the related words suggested by our system are more scientific than the suggested terms of *Wordy*. We believe this is because of the inclination of Wikipedia authors. Furthermore, all of the relevant words that are suggested by *Wordy* are also detected by our system but are ranked lower than 10 in he list. However, it is better to use regular metrics to better evaluate the proposed approach.

For further investigation, we studied the suggested keywords for the concept "Apple". This query is the first query that our system started with. Table 2 shows the top 5 categories assigned to the concept "apple" by the Open Source Project[2] (ODP) which is the largest, most comprehensive human-edited directory of the Web. It is constructed and maintained by a vast, global community of volunteer editors.

Table 2. Top 5 ODP Categories for the query "Apple"

Computers: Systems: Apple
Home: Cooking: Fruits and Vegetables: Apples
Computers: Emulators: Apple
Computers: Companies: Apple Inc.
Arts: Music: Bands and Artists: A: Apple, Fiona

As it is clear from the ODP categories there are five directories for the "apple" concept. These directories are categorized to three main categories: *Computer*, *Fruit* and *Music* related categories. Our system detected these categories and suggested some keywords for the concept "apple". The overall result for the concept "apple" is shown in Table 4 at appendix 2.

As it is shown Table 4, the EM clustering algorithm detected three clusters for the "apple" concept. To discriminate each cluster of the concept, we name them *APPLE_F*, *APPLE_C* and *APPLE_M*. Apparently, if a term exists in more than one cluster, it has more than one weight.

Table 4 shows that the first cluster matches the *Fruit* category. Only one of the documents (among ten) is assigned to this cluster by EM algorithm. The second cluster completely matches the *Computer* category and shows the most related words to the concept "apple" in the computer domain. Five documents are assigned to this cluster and the four remaining documents are assigned to the third cluster. However the distribution of the suggested words in this cluster is not so well. This cluster contains words from three different categories, *Fruits*, *Computer* and *Music*.

[2] http://www.dmoz.org

As in this research we want to investigate the usage of *recursive vector creation* method for *keyword suggestion* and not categories, so the words are much more important than their clusters for us. However, we believe it is possible to have better clustering using some methods such as applying LSA before clustering documents or increasing the number of instances (e.g. top 100 documents) to provide the clustering method with more information of each category.

5. CONCLUSION AND FUTURE WORKS

In this research we proposed an efficient and effective architecture for automatic concept graph creation. This approach is a statistical approach so it is language independent, also it does not need much processing resources. The collection that we used as the source of the system knowledge was Wikipedia collection because of its rich content. The process of concept graph construction started with a random query term and tries to find concepts that are highly related to the query term. This process is a two-step and recursive process. As an evaluation we compared the result of the system with the W*ordy* system. All of the keyword suggested by *Wordy* as top 10 keywords has been detected by our system; furthermore our system suggested some more relevant keywords in our benchmark. In future we want to use this method for semantic query expansion and retrieval purposes.

6. REFERENCES

[1] [1] V.Abhishek, Keyword Generation for Search Engine Advertising using Semantic Similarity between Terms. WWW2007. 2007.

[2] [2] Sowa, John F. "Concept Graphs for a Data Base Interface", IBM Journal of Research and Development 20(4), 336–357, July 1976.

[3] [3] Huang, W. C., Trotman, A., & Geva, S. (2007). Collaborative knowledge management: Evaluation of automated link discovery in the Wikipedia. In Proceedings of the SIGIR 2007 Workshop on Focused Retrieval, 9-16, 2007.

[4] [4]A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. ACM Comput. Surv., 31(3):264–323, 1999.

[5] [5] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620, 1975.

[6] [6] Xuezhi Zheng; Zhihua Cai; Qu Li An Experimental Comparison of Three Kinds of Clustering Algorithms. International Conference on Neural Networks and Brain, 2005.Volume 2, Page(s): 767 – 771, 2005.

[7] [7] Ian H. Witten and Eibe Frank, "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[8] [8] INEX 2006 Wikipedia Collection, http://inex.is.informatik.uni-duisburg.de/2006/. Retrieved on Sep. ,2006.

[9] [9] H. Amiri, A. AleAhmad, F. Oroumchian, C. Lucas, and M. Rahgozar. Using owa fuzzy operator to merge retrieval system results. The Second Workshop on Computational Approaches to Arabic Script-based Languages, LSA 2007 Linguistic Institute, Stanford University, USA, 2007.

[10] [10] J. Callan, "Distributed Information Retrieval," In Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval, W. Bruce Croft, ed. Kluwer Academic Publishers, pp. 127-150, 2000.

[11] [11] Adriani, M., van Rijsbergen, C.J.: Term Similarity-Based Query Expansion for Cross-Language Information Retrieval. In Lecture Notes in Computer Science, 1696, 1999.

[12] [12] Lee, M. Kang, E.-K. Gatton, T. M. Web-Document Filtering Using Concept Graph. Lecture Notes in Computer Science 2006.

[13] [13] R. R. Yager, V. Kreinovich, "Main Ideas Behind OWA Lead to a Universal and Optimal Approximation Scheme", Technical Report UTEP-CS-02-16, 2002.

Appendix 1.

Table 3. Keywords Suggested by Wordy and Our System for Queries: Skin, Teeth, Massage, Medical

Query	Wordy	Our System	Description	Weight
Skin	Skincare	Psoriasis	Chronic skin disease characterized by scaly red patches on the skin	0.998
	Facial	Inhale		0.944
	Treatment	Epidermis	Epidermis is the outermost layer of the skin	0.939
	Face	Uvb	Radiant component of sunlight which causes sunburn and skin cancer	0.938
	Care	Danger		0.937
	Occitane	Corneum	The outermost layer of the skin	0.935
	Product	Melanocytic	A small, dark spot on human skin	0.935
	Exfoliator	Harm		0.923
	Dermal	Exposure		0.916
	Body	Prolong	Skin transplantation	0.893
Teeth	Tooth	Tooth		0.999
	Whitening	Xtract		0.711
	Dentist	Dentition		0.416
	Veneer	Dentist		0.376
	Filling	Orthodontic		0.310
	Gums	Enamel		0.286
	Face	Incisor		0.246
	Baby	Dental		0.240
	Smilesbaltimore	Premolar		0.235
	Features	Molar		0.217
Massage	Therapy	Heritage		0.999
	Bodywork	Therapist		0.998
	Massageandspalv	Knead		0.998
	Therapist	Parlor		0.995
	Therapeutic	Kahuna	an expert in herbal medicine	0.953
	Thai	Erotic		0.903
	Oil	Reflexology		0.896
	Bath	Perineal		0.869
	Offer	Therapy		0.736
	Styles	Shiatsu	Japanese massage technique in which pressure is applied to specific areas of the body	0.512
Medical	Doctor	Specialist		0.998
	Clinic	Health		0.980
	Health	Maternity		0.968
	Medicine	Care		0.960

Query	Wordy	Our System	Description	Weight
	Service	Pusat	Hospital	0.959
	Offers	Hospital		0.855
	Advice	Medicine		0.676
	Search	Islam		0.669
	Member	Clinic		0.650
	Information	Practice		0.523

Appendix 2.

Table 4 shows the overall keywords suggested by our system for the concept "apple". The first column, CID, is the cluster identification number. The second column, Selected Term, shows the word suggested by the system as a related word to the concept "apple". The third column, Weight, is the degree of association relationship between the query and the selected word to the concept "apple". The words in each cluster are sorted according to their descending order of similarity to the concept "apple". To have a better understanding we have added the description column which explains the relation between the query and selected words. As description column shows all the words in the table are strongly related to the concept "apple". This high relationship between the selected words shows that the proposed system is appropriate for keyword suggestion and query expansion.

Table 4. Overall Keywords Suggested by Our System for the Concept "apple"

CID	Selected Term	Weight	Description	CID	Selected Term	Weight	Description
1[&]	pear	1.00		2	Os	0.70	
1	pie	1.00		2	truetype	0.70	
1	cinnamon	0.97	Spice made from the bark of a tree	2	gs	0.69	Type of apple computers that contains Graphics and Sound
1	pastry	0.97		2	power	0.67	
1	tart	0.96		2	intel	0.66	
1	bramley	0.93	Type of large English apple	2	mac	0.61	
1	quince	0.92		2	unicode	0.55	
1	motherhood	0.91		2	powerbook	0.55	Series of Macintosh portable computer
1	fruit	0.89		2	ii	0.51	Apple II series
1	strudel	0.89		3[#]	windows	1.00	
1	tatin	0.85	pastry	3	imac	1.00	
1	tart	0.83		3	malus	1.00	Apple Tree
1	apfelstrudel	0.79	pastry	3	wozniak	0.99	Steve Wozniak, one of the two founders of the Apple company
1	apfel	0.78	A kind of apple	3	brion	0.97	singer
1	recipe	0.60		3	record	0.94	
2[+]	windows	1.00		3	elizondo	0.91	Music producer
2	linux	1.00		3	system	0.90	
2	desktop	0.99		3	steve	0.90	Steve Wozniak
2	disk	0.97		3	video	0.90	
2	rom	0.97		3	orchard	0.89	group of planted fruit trees
2	floppy	0.97		3	ii	0.86	Apple II series
2	garamond	0.95	font designed by apple comp.	3	pollination	0.81	process of fertilizing plants
2	palett	0.93		3	fruit	0.80	
2	color	0.91		3	OS	0.80	
2	iie	0.90	Apple II series	3	processor	0.78	
2	application	0.90		3	Job	0.77	

CID	Selected Term	Weight	Description	CID	Selected Term	Weight	Description
2	subpixel	0.87		3	display	0.77	
2	redhat	0.87		3	commercial	0.77	
2	typeface	0.84		3	store	0.74	
2	glyph	0.83		3	fiona	0.72	Singer
2	display	0.77		3	cultivar	0.69	cultivated plant
2	system	0.77		3	mac	0.64	
2	sun	0.77		3	power	0.63	
2	adobe	0.76		3	g5	0.56	Apple G series
2	processor	0.72		3	macintosh	0.55	
2	iic	0.71	Apple II series	3	ipod	0.53	
2	macintosh	0.71		3	itune	0.50	software

[&] *APPLE_F*
[+] *APPLE_C*
[#] *APPLE_M*

Annotation of Scientific Summaries for Information Retrieval

Fidelia Ibekwe-SanJuan[1], **Silvia Fernandez**[2], **Eric SanJuan**[2], **Eric Charton**[2]
[1] ELICO - University of Lyon3, France. {ibekwe@univ-lyon3.fr}
[2]LIA – University of Avignon
[2] {silvia.fernandez@univ-avignon.fr, eric.sanjuan@univ-avignon.fr, eric.charton@univ-avignon.fr}

Abstract.
We present a methodology combining surface NLP and Machine Learning techniques for ranking asbtracts and generating summaries based on annotated corpora. The corpora were annotated with meta-semantic tags indicating the category of information a sentence is bearing (objective, findings, newthing, hypothesis, conclusion, future work, related work). The annotated corpus is fed into an automatic summarizer for query-oriented abstract ranking and multi-abstract summarization. To adapt the summarizer to these two tasks, two novel weighting functions were devised in order to take into account the distribution of the tags in the corpus. Results, although still preliminary, are encouraging us to pursue this line of work and find better ways of building IR systems that can take into account semantic annotations in a corpus.

Keywords. Corpus annotation, discourse structure analysis, automatic summarization, document ranking, term weighting.

1. Introduction

The question of assisting information seekers in locating a specific category (facet) of information has rarely been addressed in the IR community due to the inherent difficulty of such a task. Indeed, efficiency and effectiveness have been the main guiding principles in building IR models and tools. Our aim here is to delve into the problem of how to assist a researcher or a specialist in rapidly accessing a specific category or class of information in scientific texts. For this, we need annotated corpora where relevant sentences are marked up with the type of information they are purportedly carrying. We identified eight categories of information in abstracts which can be useful in the framework of information-category driven IR: OBJECTIVE, RESULT, NEWTHING, HYPOTHESIS, FINDINGS, RELATED WORK, CONCLUSION, FUTUREWORK. These categories enable the user to identify what a paper is all about and what the contribution of the author is to his/her field. We adopted a surface linguistic analysis using lexico-syntactic patterns that are generic to a given language and rely on surface cues to perform sentence annotation from scientific abstracts. Once annotated, the corpus is fed into an automatic summarizer which takes into account the different semantic annotations for query-oriented document ranking and automatic summarization. The automatic summarizer used here is Enertex developed by LIA team at the University of Avignon (Fernández *et al*, 2007a). Enertex is based on neural networks (NN), inspired by statistical physics, to study fundamental problems in Natural Language Processing, like automatic summarization and topic segmentation.

In this paper, we will present some preliminary experiments on abstract ranking and automatic summarization using the semantic annotations resulting from our sentence categorization scheme.
The plan of this paper is as follows: section 2 recalls relevant related work; section 3 describes the sentence categorization method. Section 4 describes the query-oriented abstract ranking and automatic summarization experiments using the semantic annotations. Section 5 discusses difficulties inherent in this task as well as earlier unsuccessful experiments which we had attempted.

2. Related Work

Of a multi-disciplinary nature, our research draws from at least two distinct research communities: NLP and IR. Our survey will thus touch on relevant work from these two communities.

There is a large body of work in the NLP community on the structure of scientific discourse (Luhn 1958, Swales 1990, Paice 1993, Salager-Meyer 1990). Following a survey of earlier works, Teufel & Moens (2002) established that scientific writing can be seen as a problem-solving activity. Authors need to convince their colleagues of the validity of their research, hence they make use of rhetorical cues via some recurrent patterns (Swales 1990[1], Teufel & Moens 2002). According to Toefel & Moens (2002), meta-discourse patterns are found in

[1] *« researchers like Swales (1990) have long claimed that there is a strong social aspect to science, because the success of a researcher is correlated with her ability to convince the field of the quality of her work and the validity of her arguments*», cited in

almost every 15 words in scientific texts. It is thus feasible to present important information from sentences along these dimensions which are almost always present in any scientific writing: research goal, methods/solutions, results. Earlier studies also established that the experimental sciences respected more these rhetorical divisions in writing than the social sciences and more often than not, used cues to announce them. One of the goals of these studies has been and continues to be automatic summarization. Discourse structure analysis is a means of identifying the role of each sentence and thus of selecting important sentences to form an abstract. Teufel (1999), Teufel & Moens (2002), and then Orasan (2001) have pursued this line of research. Patterns revealing the rhetorical divisions are frequent in full texts but are also found in abstracts. For instance, within the division « *Motivation/objective/aim* », one could find the sentence containing the lexico-syntactic cue « *In this paper, we show that...* ». Teufel & Moens (2002) showed that authors took great pains in abstracts to indicate intellectual attribution (references to earlier own work or that of other authors). Since abstracts contain only the essential points of a paper, it is to be hoped that only important sentences are there and that therefore their classification is an easier task than classifying sentences from full texts. However, abstracts will not carry all the patterns announcing the different rhetorical divisions. While categories like objective, methods and results will almost always be present, others like "*new things, hypothesis, related_work, future_work*" may be missing.

Research on automatic summarization *per se* has become very dynamic of late. Sparked off by Luhn in the late 50's (Luhn 1958) who developed a system of sentence extraction, automatic summarization is the process that transforms a source text into a target, smaller text in which relevant information is condensed. Different techniques have been explored for this task. They can roughly be split into two broad families: those relying primarily on NLP and those relying primarily on statistical / machine learning models. Quite often, a combination of techniques from the two families is necessary to produce satisfactory summaries. The dominant approach to remains automatic summarization by sentence selection rather than by real abstraction, using statistical models to rank sentences according their relevance (Mayburi Mani, 1999). Some post-processing using NLP techniques is usually needed to smoothen the most glaring coherence problems.

The works of Teufel & Moens (2002) and Orasan 2001 can be classified in the NLP-oriented approach. Teufel & Moens (2002) developed a system called Argumentative Zoner for detecting the rhetoric function of sentences according to a detailed classification of rhetoric patterns in English. They trained a Naïve Bayes classifier to categorize sentences in 80 full text scientific articles from the computational linguistics field. This classifier attained an accuracy of (73%) in classifying sentences according to the different categories of information they announced. Basing on the work of Teufel & Moens (2002), Genoves *et al*. (2007) developed the AZEA authoring tool (Argumentative Zoning for English Abstracts) to identify the discourse structure of scientific abstracts. These authors also used machine learning techniques (decision trees, Naïves Bayes, rule learning algorithm and SVM) to categorize sentences from 74 abstracts from the pharmacology domain. The SVM classifier attained the highest degree of accuracy (80.3%) on well structured abstracts. This performance dropped to 74.8% when abstracts written by learners (students) were considered.

The majority of automatic summarization systems are based on statistical and/or machine learning models. Among the criteria and techniques explored, we can cite textual position (Edmundson 1969; Brandow *et al*. 1995; Lin and Hovy 1997), Bayesian models (Kupiec *et al*., 1995), SVM (Mani and Bloedorn, 1998; Kupiec *et al.*, 1995), maximum marginal relevance (Goldstein *et al.*, 1999). These studies also take into account structural information from the document such as benchmark words and structural indicators (Edmundson, 1969; Paice 1990), a combination of information retrieval and text generation to find patterns or lexical strings in the text (Barzilay and Elhadad, 1997; Stairmand, 1996). Automatic summarization systems can also be viewed alongside the number of documents summarized at a time: single or multi-documents. Lately, the focus has been on multi-document summarization. However, at least three challenges face multi-document summarization: redundancy removal, novelty detection and detection of contradictory information. The first two problems are of course related. For the elimination of redundancy, current studies rely on temporal cues in documents. A general method for addressing novelty detection lies in extracting the temporal labels such as dates, past periods or temporal expressions (Mani and Wilson 2000) or in building an automatic chronology from the literature (Swan and Allan, 2000). Another technique that uses the well-known position of $\chi 2$ (Manning and Schütze, 1999) is used to extract unusual words and phrases from the documents. A study comparing redundancy removal techniques (Newman *et al*., 2004) showed that a similarity measure like the cosine measure between sentences attained a similar performance to other more complex methods such as Latent Semantic Indexing (LSI) (Deerwester *et al.,* 1990). Research on automatic summarization has come a long way since its beginning. Despite the residual problem of lack of coherence and cohesion, the summaries proposed by automatic systems are an approximation of the human summary.

Teufel & Moens (2002: p. 413).

Our approach to sentence categorization and query-oriented abstract ranking and summarization combines the two major techniques from the NLP and machine learning communities. We first perform sentence categorization by building on earlier works on the discourse structure of scientific texts (Teufel & Moens 2002). Like in Teufel[2] (1999), we adopt a domain-independent level of linguistic analysis. The major goal of these authors was to build summaries in such a way that the new contribution of the source article can be situated with regard to earlier works. This is in line with the recent task of "novelty detection" in multi-document summarization which was added in the last "Document Understanding Conferences" (DUC[3]) challenge. After annotating the corpus with the different categories of information each sentence contains, we perform query-oriented abstract ranking and automatic summarization. This part is done with the Enertex system, an automatic summarizer based on the neural networks approach inspired by the statistical physics of magnetic systems. Enertex is based on the concept of «textual energy». The principal idea behind Enertex is that a document can be viewed as a set of interactive units (the words) where each unit is affected by the field created by the others. The algorithm models documents as neural network whose interaction or "textual energy" is studied. Because of the nature of the links that the measure of energy induced, it connects to both sentences with common words and sentences that are in the same vicinity without sharing necessarily the same vocabulary. Textual energy has been used as document similarity measure in NLP applications. What makes this system more interesting is its ability to handle quite different tasks. In principle, the textual energy can be used to score sentences in a document and separate those that are relevant from those that are not. This led immediately to a strategy of single-document summarization by extracting phrases (Fernández *et al.,* 2007a). On the other hand, using a query as an external field in interaction with a multi-document corpus, we have broadened the scope of this idea to develop an algorithm for query-guided summaries (Fernández *et al.,* 2007b). So we calculated the degree of relevance (the textual energy) of the corpus sentences to the query. Query-guided summaries have been evaluated in the context of DUC's tasks. Enertex system compares very favorably to the other participating systems because, in essence, textual energy is expressed as a simple product matrix. Another less obvious application, is to use the information of this energy (seen as a spectrum of the sentence) and compare it with others. This allows the detection of thematic boundaries in a document. For this comparison we used the test match between Kendall. Enertex attained performances equivalent to state of the art (Fernández *et al.,* 2007a). Here, we have adapted it to the task of query-oriented abstract ranking taking into account semantic annotations present in the corpus and in the queries.

3. Lexico-syntactic patterns acquisition for sentence categorization

3.1 Corpora

To determine the type of information carried by each sentence, we need to identify and characterise the patterns that introduce that particular information type. We have selected eight categories of information which a user can seek for in scientific discourse in the framework of novelty detection: objective, results, newthings, findings, hypothesis, future work, related work, conclusions. To acquire patterns reflecting the eight categories of information we want to mark up, we analyzed corpora from three different disciplines. The 1st corpus was made up of 50 abstracts on Quantitative biology from the Open Archives Initiative (OAI[4]) containing the word '*gene'*. We manually read and analyzed the first 50 abstracts in order to formulate our initial set of patterns, seen as the seed patterns. The seed patterns were then automatically projected onto two other corpora using Unitex linguistic toolbox, in order to test their portability and to acquire new patterns. Thus, pattern acquisition was done incrementally. The second corpus consisted of 1000 titles and abstracts from 16 Information Retrieval journals downloaded from the PASCAL[5] database. The third corpus from the field of Astronomy, was made up of 1293 titles and abstracts from the ISI[6] Web of Science (WoS) database, containing the the term "Sloan Digital Sky Survey" (SDSS[7]). We describe below in more details how the initial set of patterns acquired manually from the first corpus were implemented and projected onto the two remaining corpora.

---

2 He spoke of steering "*clear of distinctions that are too domain specific*", adding that it was necessary to take into account "*robustness requirements of our approach, we cannot go indefinitely deep: the commonalities we are looking for must still be traceable on the surface*" (ibid, p.83).

3 duc.nist.gov/guidelines/2007.html

4 http://fr.arxiv.org/archive/q-bio

5 http://www.inist.fr.

6 Institute for Scientific Information

7 http://www.sdss.org/

3.2 Implementation of the patterns as finite state automata

Lexico-syntactic patterns announcing a specific information type are not fixed expressions. They are subject to variations. These variations can occur at different linguistic levels: morphological (gender, number, spelling, inflection), syntactic (active/passive voice, nominal compounding *vs* verbal phrase), lexical (derived form of the same lemma) and semantic (use of synonymous words). The exact surface form of all these variations cannot be known in advance. Hence, categorizing sentences based on these surface patterns requires that we take into account places where variations can occur so as to ensure that they can be applied to new corpora with a certain degree of success. From our manual study of the 50 abstracts in Quantitative biology, we wrote contextual rules in the form of regular expressions implemented as finite state automata in the Unitex[8] system. These automata were then projected on the two test corpora to identify the different categories of sentences. Verbs are searched for in their infinitive form, nouns in their noun masculine gender.

Figure 1 below shows the finite state automaton that recognize OBJECTIVE sentences.

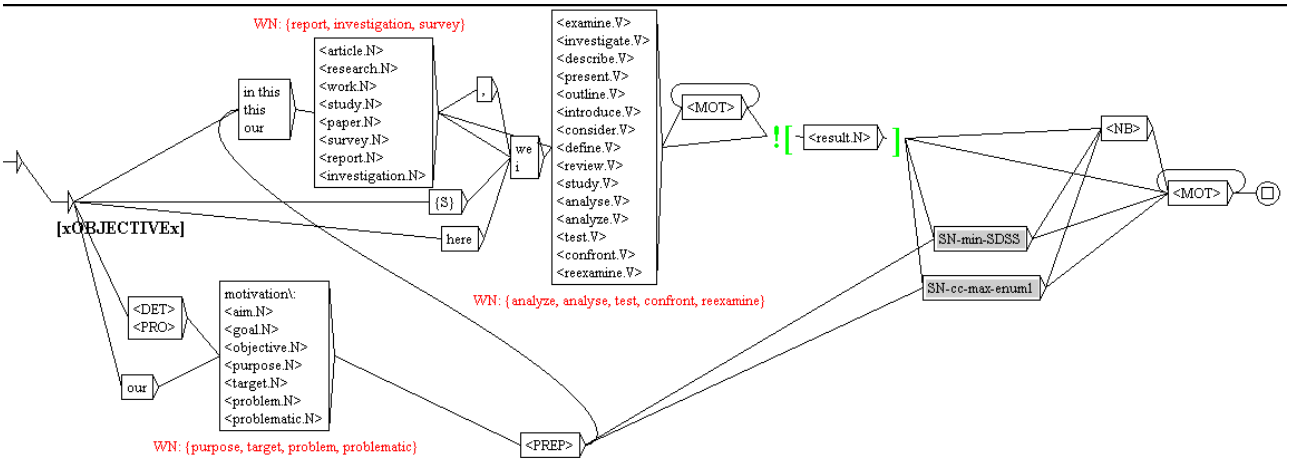

Figure 1. Finite state transducer that categorizes sentences as «OBJECTIVE».

The recognition of the patterns require the combined use of POS[9] and lexical information and syntactic-level information (recognition of noun phrases in the context of a lexical pattern). In Figure 1, the pattern for identifying objective sentences contains a path which searches for a sentence with a determiner (<DET>) or a pronoun (<PRO>), followed by words like "*goal, objective, purpose, aim,...*" then by a preposition (<PREP>), and by a noun phrase (SN-cc-max-enum1). The grey boxes call other finite state grammar embedded in the current one. For instance, «SN-cc-max-enum1» is a local grammar that identifies complex NPs (NPs with embedded simpler NPs). This grammar in turn, embeds another simpler NP grammar. The expressive power of such local grammars can be quite high as simpler grammars are embedded into more complex ones to achieve a considerable level of complexity. Each category of information is represented by a single automaton with multiple paths. Note however that some lexico-syntactic patterns are ambiguous and can introduce two different categories of information. For instance, there is not always a clear boundary between patterns announcing the objective of a paper and its results. "*In this paper we show that...*" could announce either "*objective"* or "*results*". Genoves *et al*. (2007) observed that the classifiers they trained could not distinguish properly between "methodology" and "results" patterns.

To ensure the completeness of our lexico-syntactic patterns and hence their portability on other domains, we expanded the lexical lists in the patterns with words in the semantic equivalence classes from an external lexical database, in this case WordNet[10]. However, WordNet being a general vocabulary semantic resource, has every conceivable sense for a given word, some of which were not appropriate for scientific writing. For instance, the verb "*show*" has among its synsets the following "*render* (sense of picture), *read*, *register*, *evince*" which are senses rarely encountered in scientific writing. A second unwelcome phenomenon in expanding word lists with WordNet is that if word w_0 has as synonyms word w_1, there is no guarantee that the synonyms of word w_1 will be synonyms of word w_0. In other words, synonymy is neither always symmetric nor transitive. For instance, among

[8] www-igm.univ-mlv.fr/~unitex/

[9] Part-Of-Speech

[10] http://wordnet.princeton.edu/perl/webwn

the synonyms of "*obtain"*, is the word "*receive"*, the latter has synonyms like "*welcome, meet, pick up"* which are clearly not synonyms "*obtain*" in the sense used in scientific articles. Of the total of 9506 sentences in the SDSS corpus, 1 882 (19%) unique sentences were tagged by our restricted patterns and 1 959 (20%) sentences by the expanded patterns with WordNet, thus the coverage by adding lexical entries (synonyms) from an external resource was not significantly increased.

3.3 Corpus annotation

Once the patterns have been built and tested, the second stage is to mark-up sentences in the corpus with the category of information they announce. This is done by using the transducer option in Unitex. Transducers are variants of the grammars that modify the text by performing a re-writing operation such as "insert, delete, copy". The information carried by each pattern is inserted at the beginning of the sentence containing the pattern. Figure 2 shows an example of the output by the transducers of each local grammar. The tags [OBJECTIVE, RESULT, HYPOTHESIS] were inserted by our finite state grammars.

{S}ISI:{S}000240201200022.
{S}Potential sources of contamination to weak lensing measurements: constraints from N-body simulations.
[OBJECTIVE] {S}We investigate the expected correlation between the weak gravitational shear of distant galaxies and the orientation of foreground galaxies, through the use of numerical simulations.{S} [HYPOTHESIS] This shear-ellipticity correlation can mimic a cosmological weak lensing signal, and is potentially the limiting physical systematic effect for cosmology with future high-precision weak lensing surveys.{S} We find that, if uncorrected, the shear-ellipticity correlation could contribute up to 10 per cent of the weak lensing signal on scales up to 20 arcmin, for lensing surveys with a median depth z(m) = 1.{S} The most massive foreground galaxies are expected to cause the largest correlations, a result also seen in the Sloan Digital Sky Survey.{S} [RESULT] We find that the redshift dependence of the effect is proportional to the lensing efficiency of the foreground, and this offers prospects for removal to high precision, although with some model dependence.{S} The contamination is characterized by a weakly negative B mode, which can be used as a diagnostic of systematic errors.{S} We also provide more accurate predictions for a second potential source of error, the intrinsic alignment of nearby galaxies.{S} This source of contamination is less important, however, as it can be easily removed with distance information.

Figure 2. Example of an annotated abstract.

Next, we evaluated the accuracy of our sentence tagging grammars by manually verifying the output of the different automata on the SDSS corpus. Each sentence was read in order to ascertain if it really belonged to that particular category of information. The table below gives figures on the accuracy of the each automaton in annotating sentences from a given category. The 2nd column is the total number of sentences tagged by an automaton. The 3rd column gives the ratio of correctly tagged sentences over all tagged sentences (precision). The 4th column is the proportion of errors amongst sentences tagged. Recall could not be measured because we could not read the entire corpus to exhaustively identify all the sentences belonging to a specific category that were not tagged. In the future, we plan to measure recall on a sample of the corpus. The automaton for hypothesis sentences embeds the one for "finding" because the two categories of information are often announced by similar patterns. This explains why we have seven patterns in the table instead of the eight announced previously.

Pattern	**Occ.**	**Prec.**	**Errors**
RESULT	500	100%	0
CONCLUSION	206	193 (94%)	13 (6%)
FUTURE_WORK	198	194 (98%)	4 (2%)
NEWTHING	505	485 (96%)	20 (4%)
OBJECTIVE	513	513 (100%)	0
RELATED_WORK	31	30 (97%)	1 (3%)
HYPOTHESIS	487	479 (98.4%)	8 (1.6%)

Table 1. Accuracy measure of the automata for tagging sentences on the SDSS corpus.

As we can see, our patterns achieved a high level of accuracy in tagging sentences with the correct type of information (> 94%). The majority of the errors observed in the conclusion sentences came from the fact that the word "conclusion" or "conclude" which are triggers for tagging a sentence as such were present in the sentence

but the actual conclusion came in the following sentences (see appendix A for examples). A possible way of correcting this would be to extend the conclusion class to the "*n*" sentences following the one containing that word. The majority of the errors observed in the hypothesis-findings categories come from recommendations using the trigger word "*should*" or from future work using the word "*shall*". Positive and negative examples of sentences tagged for each category of information can be found in the appendix. For two categories of patterns (objective, result), we could find no error in the tagged sentences. This might be due to the highly technical nature of the SDSS corpus. We might observe more errors in a less technical corpus.

3.4. Automatic pattern generation

A limit of the rule-based approach which we have adopted here for pattern acquisition and sentence tagging is that it is impossible to capture all the potential patterns especially in unseen texts. Previous studies used machine learning techniques to address this issue. Teufel & Moens 2002, then later Genoves *et al*., 2007 trained classifiers on manually hand-crafted patterns. However, the authors trained the classifiers on the same corpus as the initial one used to build the patterns in order to evaluate their accuracy. They did not actually use them to learn new patterns.

As a first step to new patterns acquisition, we applied a rule generator in order to systematically generate all the possible lexical combinations of words in similar contexts in the patterns in the eight categories. Here is a detailed description of the algorithm.

To find the generated patterns, we use a substitution class. Consider the sample class of substitution called « *demonstrate* » from the result category (left box in figure 3 below). Our algorithm will first locate all the occurrences of each term of the « *demonstrate* » class in the corpus, and associates them with *n words* before or *n words* after. For example with n=2, and only word after, ee could find in the corpus the patterns in the middle box. For the purpose of this presentation, let us call those extended patterns *P1*. In a second step, we will substitute in P1, all the terms of the *demonstrate* class. This will give us a new list of candidate patterns. If we achieved this on P1 by substitution of the « *demonstrate* » class, we could have a proposition list, called P1' (the rightmost box).

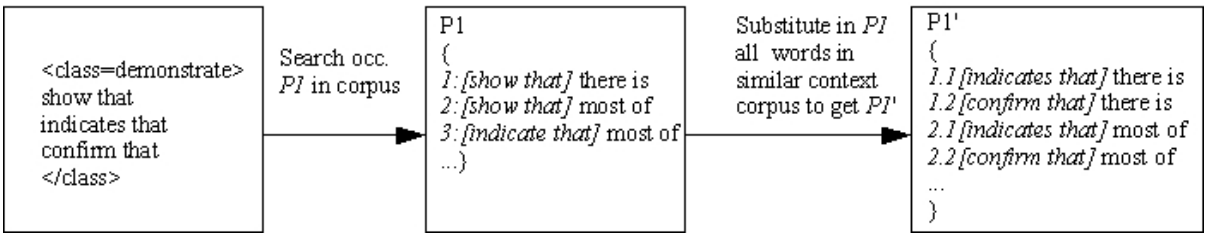

Figure 3. Flowchart of the pattern generator.

We see in P1', that 1.1 and 1.2 are substitutions of the *demonstrate* class terms, applied on the sequence 1 from P1. The rule generator checks for the presence of these new patterns in the corpus and records their occurrences. This ensures that our sentence classification program will not miss any sentence carrying a specific information type owing to lexical variations by substitution of one synonymous word with another. Owing to lack of time, we were not able to apply the generated patterns on the SDSS corpus in order to evaluate the accuracy and number of annotated sentences. This will be the object of future research.

4. Query-oriented abstract ranking

In this section, we explore how the semantic tags inserted in the abstracts (cf. §3.2 - 3.3) can be used for query-oriented abstract ranking and multi-abstract summarization. Enertex was selected as an appropriate tool because of its ability in capturing non-direct relations between queries and abstracts. We implemented a new combination of weighting functions in the Enertex system specifically for these tasks. One of the additional advantages of this system is its ability to handle quite different tasks of text selection and ranking with minor changes. First, we give a general description of the system.

4.1. Text representation in the Enertex system

The system builds large matrices *M* of word occurrence in a collection of small texts and computes a similarity between texts based on $(M.M^t)^2$. This is the matrix representation of the energy in the magnetic Ising

model (Fernández *et al*, 2007a). Documents are pre-processed following conventional methods. First, functional, stop words and numbers are filtered out. Then normalization and lemmatization of words are carried out to reduce the dimensionality. A *bag-of-word* representation of the texts is performed, yielding a matrix $M=[f(w,s)]_{w\in W, s\in S}$ of weighted frequencies consisting of a set S of P sentences (lines) and a vocabulary W of $i = 1, \cdots, N$ terms (columns), where f is a weighting function on pairs of words and sentences. We use some elementary notions of the graph theory to describe Enertex approach.

Let us consider the sentences as sets S of words. These sets constitute the vertices of the graph. We draw an edge between two vertices s,t every time they share at least a word in common. We obtain the graph $I(S)$ from intersection of the sentences. We weigh these pairs $\{s, t\}$ which we call edges using the weighting binary function f on pairs of words and sentences :

$$e(s,t)=\sum_{w\in s\cap t}(f(w,s)\times f(w,t)) \qquad \text{[Equation 1]}$$

For automatic summary, $f(u,s) = 1$ if u is in s, and 0 otherwise. In this special case, $e(s,t)$ is the number $|\, s \cap t \,|$ of words that share the two connected vertices. Finally, we add to each vertex s an edge of reflexivity $\{s\}$ valued by the sum of weights $f(u,s)$ of words u in sentence s.
This weighted intersection graph is isomorphic with the adjacency graph $G(M \times M^T)$ of the square matrix $M \times M^T$. In fact, $G(M \times M^T)$ contains P vertices. There is an edge between two vertices s,t if and only if $[M \times M^T]_{s,t} > 0$.
The matrix of Textual Energy E is $(M\times M^T)^2$. This matrix is computed using its adjacent graph whose vertices are the same as those of the intersection graph $I(S)$ and:

- there is an edge between two vertices each time that there is a path of length 2 in the intersection graph;
- the value of an edge: a) loop on a vertex s is the sum of the squares of the values of adjacent edges at the vertex, and b) between two distinct adjacent vertices r and t is the sum of the products of the values of the edges on any way with length 2 between both vertices. These ways can include loops.

From this representation, it can be seen that the matrix of Textual Energy connects at the same time sentences sharing common words because it includes the intersection graph as well as sentences in the same neighborhood but not necessarily sharing the same vocabulary. Thus, two sentences s, t not sharing any word in common but for which there is at least one third phrase r will be connected all the same. The strength of this link depends in the first place on the number of sentences in its common neighborhood, and on the vocabulary appearing in a common context. This constitutes the main distinction with other usual similarity measures like cosine or mutual information measures that are based on direct co-occurrences of terms. Therefore, Textual Energy is comparable to Latent Semantic Indexing (LSI) without requiring the expensive computation of the Singular Value matrix decomposition. The advantage of such a model is that they allow to directly use query terms that appear only once in the corpus but that are closely related to some central topics. Since Textual Energy is based on a simple graph model, it is more adaptable to different applications. In text summarization, it was tested on several corpus including DUC 2006 and DUC 2007. Its performance was measured with ROUGE metrics and it showed similar performances to other state of the art systems (Fernandez *et al.*, 2007b).

4.2. Ranking abstracts with semantic annotations

We now describe how Enertex was adapted to the task of ranking abstracts. All the experiments here were performed on the SDSS corpus. First, each abstract is considered as a unique *bag-of-words* (a sentence). In other to take into account the frequency of words in abstracts and to favor low frequency words that best characterize an abstract, we used the following weighting function f on pairs of words and sentences based on the so-called "equivalence index" which is the product of the conditional probabilities P(s/w) and P(w/s). Only values over a threshold of the form 10^{-n} where n depends on the corpus size are considered. Thus we set:

$$f(w,s)=\log(trunc((\frac{f_{w,s}^2}{f_{w,.}\times f_{.,s}}>10^{-n})\times 10^n)) \qquad \text{[Equation 2]}$$

where $f_{w,s}$ is the absolute frequency of word w in s, $f_{w,.}$ is the frequency of w in the corpus and $f_{.,s}$ is the number of words in sentence s. To optimize the ranking algorithm, we truncated float numbers to work only on integers and we cut too big values using the log. We tested the common versions of TF.IDF measures but due to double matrix product involved in the calculation of the Textual Energy matrix, the results showed an exaggerated effect of any weighting on the S matrix favoring tacitly the extreme cases (long or short phrases; frequent or infrequent terms).
Sentences are then ranked based on their weighted degree in the adjacent graph of $(M\times M^T)^2$: the score of a sentence s is set to the sum of $E_{s,t}$ for any sentence t.

The system selects the most representative abstracts and displays them in chronological (by publication date). If two abstracts have the same score, only the first one by chronological order is displayed.
When ranking abstracts in response to a query, the query *q* is considered as an abstract itself. The corpus of abstracts is ranked according to the $E_{s,q}$ value. If the query contains a very general word then the ranking is similar to the one obtained without query.

By way of example, let us consider the query: "*Randall-Sundrum"*. This term is the name of a space geometry model which occurred only once in the SDSS corpus. Using the above defined weighting function, Enertex ranked the abstract containing "*Randall- Sundrum"* and those dealing with geometry models. Enertex found the relationship between the named entity in the query and the geometry models based on the context in which it found the query term. Examples of relevant terms in this context are *geometry, spatially flat, dimension, inflation, expansion, brane, braneworld, DGP model*. This is similar to a query expansion procedure in which terms from the top ranked abstracts are used to expand the query term. The difference here is that Enertex selects the top ranked abstracts to expand the query based on the adjacent graph of the Energy matrix. Figure 4 shows one of these abstracts ranked on 7th position. Relevant terms are underlined.

Two new one-parameter tracking behavior dark energy representations omega=omega(0)/(1+z) and omega=omega(0)e(z/(1+z))/(1+z) are used to probe the geometry of the Universe and the property of dark energy. The combined [RESULT] type Ia supernova, Sloan Digital Sky Survey, and Wilkinson Microwave Anisotropy Probe data indicate that the Universe is almost spatially flat and that dark energy contributes about 72% of the matter content of the present universe. The observational data also tell us that omega(0)similar to-1. It is argued that [FINDING] the current observational data can hardly distinguish different dark energy models to the zeroth order. The transition redshift when the expansion of the Universe changed from deceleration phase to acceleration phase is around z(T)similar to 0.6 by using our one-parameter dark energy models.

Figure 4. One of the top ranked abstracts for the query "*Randall-Sundrum".*

Figure 4. Astract ranked 7th for the query "*Randall-Sundrum".*

If now we want Enertex to take into account the semantic annotation inserted into the abstracts following the connections in the same adjacent graphs. A difficulty we have to deal with here is that by definition, if the summaries follow the hypothesis of well-formedness, each semantic category tag will tend to be uniformly distributed across the corpus and will therefore have a high occurrence. Thus, when considered as words, the tags are simply ignored by the weighting function. To overcome this handicap, we multiplied our weighting function by a *g* factor that measures by how much the frequency of a word is greater than the expected one. Due to the corpus size, we could not apply complex statistical tests and most of the calculus had to be done on integers. Finally, we tried the following function:

$$g(w,s)=\log\left(trunc\left(\frac{(f_{w,s}-\overline{f_{w,.}}>0)^2}{\sum_{t\in S}(f_{w,t}-\overline{f_{w,.}})^2}\times f_{w,.}\right)\right) \qquad \text{[Equation 3]}$$

This function compares the frequency of a word or a tag to the average frequency of this item in abstracts. Only items above the average are considered as index of abstracts. Therefore this function allows us to also consider some frequent tags as abstract index .We combine the two functions *f* in Equation 1 and *g* in Equation 2 by taking their product: (f(u)+1).(g(u)+1) if at least one of the two terms in not null (f(u)+f(g)>0) to obtain a ranking that both considers specialized terms in query and general tags.
For example, we added semantic tags to the previous query "*Randall-Sundrum NEWTHING FINDING"*. The results showed that this combination effectively allows the system to rank abstracts according to these two principles. Abstracts containing the query terms are still ranked first but those containing an unusual number of tags in the query are favored. Figure 5 shows some sentences of abstract ranked on 19th position. It contains at the same time terms like "*dimensional"* related with "*Randall-Sundrum"* and "*FINDING"*. Relevant terms are underlined. In the previous case, without any tag in the query, the same abstract had been ignored.

Overall, the galaxy spectral energy distribution in the entire ultraviolet to [FINDING] near-infrared range can be described as a single-parameter family with an accuracy of 0.1 mag, or better. This nearly one-dimensional distribution of galaxies in the multidimensional space of measured parameters strongly supports the [CONCLUSION] conclusion of Yip et al., based on a principal component analysis, that [FINDING] SDSS galaxy spectra can be described by a small number of eigenspectra.

Figure 5. Some sentences from an abstract ranked 19th for the query "*Randall-Sundrum NEWTHING FINDING".*

Table 2 shows the differences between these two queries according to the content of some terms related to "*Randall-Sundrum"* and *NEWTHING FINDING* tags. It presents the percentage of ranked abstract where they appear. We observe that the percentage of related terms is almost the same and the percentage of tags used in the query increases significantly.

Query	**Some terms related with** *Randall-Sundrum: geometry, spatially flat, dimension, inflation, expansion, brane, braneworld, DGP model*	**Tags:** *NEWTHING, FINDING*
Randall-Sundrum	37%	57%
Randall-Sundrum NEWTHING FINDING	30%	88%

Table 2. Percentage of ranked abstract where terms related with *Randall-Sundrum* and *NEWTHING FINDING* tags appear.

We emphasize that with this ranking function, we do not need to specify which words are tags and which are terms. The ranking function naturally distinguishes them based on their frequency. This functionality should guarantee a high stability of the resulting rankings even when the annotation of texts is incomplete. Since we used a unifying model, the system should automatically learn from the context in which existing annotations appear to process other texts that should have been annotated in a similar way. We plan to evaluate this stability property on partially enriched texts from heterogeneous sources. The final system that we target, will rely on a:

1. tagging using the finite state automata introduces in section 3.2 to partially tag texts. This tagging appears to have a high precision but since we cannot evaluate its recall, we shall consider it as partial.
2. learning process based on the automatice pattern generation introduced in section 3.4 that shall detect text features of succeeding text related to tags.
3. text analysis based on Textual Energy that can relate abstracts to queries made up of any list of terms including those appearing once in the text collection and tags relying on a precise but partial tagging.

4.3. Query-oriented multi-abstract summarization using semantic annotations

Here, we focus on query-oriented multi-abstract summarization. In this context, summaries are a selection of documents' abstracts (instead of sentences) displayed by chronological (publication) order. To better evaluate the ability of the system to capture non-direct relations between queries and abstracts, and to determine the impact of the semantic tags in the queries, we present two types of experiments. The first one involves one-word queries consisting of abbreviations or of astronomy concepts. The aim is to see if the system was capable of producing summaries that contained a definition of the abbreviation and some related information. The experiment will consist of queries with and without tags to study their impact in summary content. The second experiment consists of phrase-tagged queries describing a phenomenon or problem related to astronomy. In this case, the aim is to observe if the summary gives information that helps to explain the problem raised in the query. The different tags will be added too to give a predominant intention to the summary.

4.3.1 One-word queries

Consider a set of four one-word queries consisting of abbreviations or of astronomy concepts (Table 3), the aim was to see if the system was capable of producing summaries that contained a definition of the abbreviation and related information. Given a compression rate *r*, the system selects the top most ranked abstracts such that the total number of their words over the total corpus size is less than *r*. We fixed the compression rate of the resulting summary to <5 % of the corpus size in terms of total number of words. The SDSS corpus is made up of 258 775 words. This induces that summaries produced by the system can contain different numbers of abstracts depending on their size in words. If two abstracts have almost the same score, they are considered redundant and only the most recent is selected. If all abstracts have a null score because no one could be related to the query, the summary will be empty. Therefore, the length of the summary also depends on the number of abstracts with a non null score.

We present now a preliminary evaluation of our approach. First we check that Textual Energy is sufficient to relate queries to abstracts. Until now, Textual Energy was used to rank sentences in which a term rarely appeared twice, meanwhile here we consider abstracts. To evaluate its performance, we consider query terms with very low frequencies but that are acronyms involving relevant topics of the corpus. Based on established definitions of these acronyms, the evaluation consisted in counting the number of relevant terms in these definitions that appear

in the abstracts selected by the system. This done, we will enrich the query with tags and see if the document ranking is modified or if it fails because of the high frequency of tags.
Table 3 shows the four one-word queries, their occurrence in the corpus, the size of the generated summary in number of words. For ease of comprehension, we added the definitions of the query terms and indicated the websites from where they were taken.

Id	Query	Corpus occ.	Nb. words summary	Definition of query term
b1	ACDM	5	0	ΛCDM or Lambda-CDM is an abbreviation for Lambda-Cold Dark Matter.
b2	AGB	2	9690	Asymptotic Giant Branch. http://www.eso.org/projects/vlti/science/node8.html
b3	AMIGA	2	9679	Analysis of the interstellar Medium of Isolated GAlaxies http://amiga.iaa.es:8080/p/1-homepage.htm
b4	LBG	3	9692	Lyman break galaxies http://www.astro.ku.dk/~jfynbo/LBG.html

Table 3. Examples of one-word abbreviation queries tested and their definitions.

From the results, it appeared that for small values of n (<4) in equation 2, only terms of very low frequency (<5) are retained. Since we carried out the experiments with n=4, query b1 (*ACDM*) did not produce any abstract.
Analysing the contents of the abstracts selected to build the summary for query b2 "*AGB*", we note:
- that the summary contains 48 abstracts;
- the presence of the scientific term used in the query (*AGB*);
- the presence of scientific terms present in the query term's presentation on the website such as *Asymptotic Giant Branch* (2 occurrences in the summary), *Life* (2), *Core* (5), *Non-LTE* (4), *Convection atmosphere* (3), *stratosphere* (3), *chemical evolution* (1).

Enertex was again able to find the relationship between the named entity in the query and the related concepts based on the context in which it found the query term.
Similarly, query b3 "*AMIGA"* produced a summary of 31 abstracts. Comparing this summary with the persentation of the term on a website (address in table 3), we found the following terms in common: *amiga* (2), e*nvironment* (29), *interaction* (3), *correlation* (15), *environmental density* (1), *isolated galaxy* (7), *denser environments* (15), *wavelength* (3), *Catalog of Isolated Galaxies* (1).
Finally, query b4 (*LBG*) produced 34 abstracts that share with the website presentation the following terms: *LBG* (7), *Ultraviolet* (3), *Red* (in the rigth context: 1), *Lyman* (8), *Rest-frame* (1), *Ly-α* (2)
Another important observation is that terms specific to queries b2 and b3 like *AGB, lte* and a*miga* are not present in the summary of b3. Meanwhile more general terms relavant to b2 and b3 like "*isolated galaxy"* and "*denser environements*" occured also in the summary of b4 but with a much lower frequency (1 and 2 respectively).The summaries were obtained using the product (f(*w,s*)+1).(g(*w,s*)+1) of two formulas in Equations 2 and 3. However, the second factor (g(*w,s*)+1) did not influence the ranking, this factor being equal to 1 for all query terms *w* and all abstracts *s*.

Seeking to determine the impact of semantic tags in the query, we added the tags announcing hypothesis, findings and objectives to the queries. We observed that all produced summaries are non null, even for query b1. This is due to the effect of the second factor *g*. To illustrate this, let us take a closer look at the results produced for query b2 "*AGB*". Similar observations can be made for the other queries. Relevant terms in the presentation of b2 mentioned in table 3 are less frequent but still present. In table 4, we can see that the importance of scientific terms related to "*AGB*" that were present in the presentation mentioned in table 3 has declined but are still present. On the other hand, the total number of tags for "hypothesis, finding and objective" are higher in the selected abstracts.

Term occurrence in the summary	query without tags	query with tags
agb	2	1
life	2	1
core	5	3
non-LTE	4	0
convection atmosphere, stratosphere	3	0
chemical evolution	1	1
hypothesis	4	15
finding	8	19
objective	19	15

Table 4. Frequency of relevant terms and of tags in summary produced for query b2 "*AGB*".

4.3.2 Phrase-tagged queries

The aim here is to evaluate to what extent the summary gives information to explain the problem raised in the query and if the use of tags orients the predominant intention in the generation of the summary. An example is the query **"***NEWTHING spectral classification of quasar"*. Table 5 shows the percentage of ranked abstracts where terms related to the query and semantic annotations are present. Relevant terms appeared in all abstracts forming the summary. The tag **"***NEWTHING"* was more present than others tags.

Terms related *(luminosity, quasar, redshift, quasar spectra, spectrum, optical~, Balmer, eigen~, Fe II emission, Baldwin)*	NEWTHING	RESULT	CONCLUD	HYPOTHES	OBJECTI	FINDING
100%	72%	60%	16%	20%	48%	24%

Table 5. Percentage of ranked abstract where terms query related (first column) and tags appear for the query "*NEWTHING spectral classification of quasar"*.

Figure 6 shows some of the sentences taken from an abstract ranked 1st and 14th respectively. Terms relevant to the query are underlined.

[NEWTHING] We found that more infrared *luminous* galaxies tend to have a smaller local galaxy density, being consistent with the picture where luminous IRGs are created by the merger-interaction of galaxies that happens more often in lower-density regions.

We find strong correlations between the [NEWTHING] detection fraction at other wavelengths and optical properties such as flux, colours and emission-line strengths.

Figure 6. Some sentences of ranked abstracts for the query **"***NEWTHING spectral classification of quasar"*. Relevant terms and tags are underlined.

In another query, the phrase "*existence of the Gunn-Peterson"* was entered in combination with different semantic tags. We did an evaluation of system's effectiveness by measuring again the presence of the relevant terms in the summary. We have identified as relevant terms related with "*existence of the Gunn-Peterson"*: *neutral hydrogen, intergalactic medium, IGM, detection+existence, quasar spectra, Lyman+Alpha, z=5.99,6.28, reionization.* The results are shown in Table 6. We observe that relevant terms are always very present in the summary and the use of a tag in the query favors his presence in the final condensed.

Id	**Query**	**Terms related with** "*existence of the Gunn-Peterson"*	**HYPOTHESIS**	**FINDING**	**CONCLUD**	**RESULT**
p2	*HYPOTHESIS existence of the Gunn-Peterson*	93%	**43%**	17%	17%	35%
p3	*FINDING existence of the Gunn-Peterson*	84%	24%	**48%**	24%	56%
p4	*CONCLUSION existence of the Gunn-Peterson*	89%	18%	29%	**33%**	37%
p5	*RESULT existence of the Gunn-Peterson*	89%	22%	33%	22%	**44%**

Table 6. Query "*existence of the Gunn-Peterson"* in combination with different tags.

5. Discussion

Regarding the sentence classification task, we observed that the same patterns can announce different information categories or that two different patterns can be present in the same sentence, thus leading to multiple tags. In the following sentence, the future_work tag is triggered by the word « *future* » while the hypothesis tag is triggered by the presence of "*can*"*:*

« We assess the accuracy with which [xHYPOTHESISx] [xFUTURE_WORKx] future galaxy surveys can measure .

cosmological parameters. »

Teufel & Moens (2002) already observed the same phenomenon on a different corpus which was from computational linguistics. In this case, the sentence will belong to the two classes as there is no clear way of determining which category should take precedence.

Concerning the sentence ranking and automatic summarization tasks, we first tried to generate query-oriented multi-abstracts summaries by sentence selection. The results were not satisfactory because the extracted sentences lacked sufficient context to be coherent. Moreover, the resulting ranking of sentences was similar to a random ranking. The use of semantic tags in queries did not change this outcome as if the sentences did not contain enough informatiion to be related to queries. We next tried to use the weighting function in Equation 3 to generate summaries from DUC 2006 corpus. The generated summaries had lower quality scores for ROUGE measures than those obtained without using this weighting function. This shows that ranking abstracts is a different task from ranking full-text documents. We then tried ranking sentences from this corpus but encountered the same problem as previously.

It was then we had the idea of working at the level of abstracts. At this level, query terms can be related to similar terms appearing in the same abstracts but in different sentences. In a sentence, a word relevant to the query typically appears once whereas this is not the case in abstracts. Because the common versions of TF.IDF did not produced the expected effects, it was thus necessary to define special weighting functions that captured low frequency terms in the corpus but which are more frequent in a smaller set of abstracts. This gave rise to the function proposed in Equation 2. This formula could not take into account semantic tags that are both frequent and uniformly distributed in all abstracts. Indeed, any well written scientific abstract would tend to contain at least one category of patterns from the major rhetorical divisions (objective, method, results, conclusion). However, some abstracts can contain an unusual number of these patterns and this information could be relevant for document ranking. Equation 3 is meant to capture such unusually high frequency of rhetorical patterns in abstracts.

Finally we found out that working at the abstract level, it was possible (as described in section 4) to define weighting functions that can take into account both rare terms and frequent semantic tags considered as supplementary words in the text. This opens an avenue for research where standard IR engines could, with minor changes, be applied on annotated corpus.

Enertex was initially designed for automatic summarization by sentence extraction and text segmentation. It attained performances equivalent to state of the art summarizers and segmentation systems. Here, we have adapted it to the task of query-oriented abstract ranking taking into account semantic annotations present in the corpus and in the query. We have to pursue these experiments in oder to determine the best way of focusing the generated summaries or rankings on a specific information type. Also, we have to set up a more rigorous evaluation framework using corpora with benchmarked results such as the DUC collections. However, what makes this system most interesting is its ability to handle quite different tasks of text selection and ranking with minor changes.

This work had a double purpose. First it shows an easy way to tag peer-reviewed abstracts according to the information carried by each sentence. Second it shows how tags can be used in a text analysis process with the view to perform automatic summarization. Text analysis tasks are part of information retrieval, they rely on reduced document collections extracted from large databases using standard Information Retrieval methods but requiring a higher level of text understanding. The methods we developed in this work constitute a novel and integrated approach for addressing advanced information retrieval tasks.

.

References

1. Barzilay R., Elhadad M., (1997), Using lexical chains for Text Summarization, Proc. ACL Intelligent Scalable Text Summarization, 10–17.
2. Brandow R., Mitze K., Rau L., (1995), Automatic condensation of electronic publications by sentence selection, *Information Processing and Management 31*, 675–685.
3. Deerwester S., Dumai s S., Furnas G., Landauer T., Harshman R., (1990), Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science 41*(6), 391–407.
4. Edmundson H. P., (1969), New Methods in Automatic Extraction, *Journal of the Association for Computing Machinery 16*(2), 264–285.
5. Fernandez S., Sanjuan E., Torres-Moreno J. M., (2007a) Energie textuelle des mémoires associatives. In N. H. ET Philippe Muller, Ed., Proceedings Traitement Automatique des Langues Naturelles (TALN), Toulouse, 2007, Toulouse, 25–34.
6. Fernandez S., Sanjuan E., Torres-Moreno J. M., (2007b) Textual energy of associative memories: performants applications of enertex algorithm in text summarization and topic segmentation. In Pro. MICAI ’07, Aguascalientes (Mexico).
7. Goldstein J., Carbonell M., Kantrowitz, Mittal V., (1999), Summarizing text documents : sentence selection and evaluation metrics, Proc. 22nd ACM SIGIR Conference, 121–128. Berkeley.
8. Kupiec J., Pedersen J., Chen F., (1995), A trainable document summarizer, Proc. *18th ACM SIGIR*, 68–73, ACM Press.

9. Lin C., Hovy E., (1997), Identifying Topics by Position, Proc. ACL Applied Natural Language Processing Conference, 283–290.Washington.
10. Luhn H.P. (1958), The automatic creation of literature abstracts, *IBM Journal of Research and development,* 2(2), 159-165.
11. Genoves L., Feltrim V.D., Dayrell C., Alusio S. (2007), Automatically detecting schematic structure components of English abstracts: building a high accuracy classifier for the task, in Proc. Recent Advances in Natural Language Processing (RANLP 2007), 7p.
12. Mani I., Bloedorn E., (1998), Machine learning of generic and user-focused summarization, Proc. *AAAI'98/IAAI'98*, Menlo Park, 820–826.
13. Mani I., Maybury M., (1999), *Advances in automatic text summarization*. The MIT Press, U.S.A.
14. Manning C. D., Schütze H., (1999), *Foundations of Statistical Natural Language Processing,* Cambridge, Massachusetts: The MIT Press.
15. Mani I., Wilson G., (2000), Robust temporal processing of news. Proc. 38th Association for Computational Linguistics, Morristown, NJ, USA, 69–76.
16. Mann W., Thompson S., (1987), *Rhetorical Structure Theory: A Theory of Text Organization,* University of Southern California, Information Sciences Institute.
17. McKeown K., Radev D., (1995), Generating summaries of multiple news articles, Proc. *18th ACM SIGIR*, 74–82.
18. Newman E., Doran W., Stokes N., Carthy J., Dunnion J., (2004), Comparing redundancy removal techniques for multi-document summarisation. Actes de *STAIRS*, 223–228.
19. Orasan C. (2001), Patterns in scientific abstracts, in Proceedings of the Corpus Linguistics 2001 Conference, Lancaster University, Lancaster, UK, 2001, 433-443.
20. Orasan C. (2005), Automatic annotation of Corpora for Text Summarisation: A Comparative Study. In Proceedings of 6th International Conference, CICLing2005, Mexico City, Mexico, February, Springer-Verlag, 670 – 681.
21. Ou S., Khoo C.S, Goh D.H. (2007), Automatic multidocument summarization of research abstracts: design and user evaluation, *Journal of the American Society for Information Science and Technology*, 2007, 58(10) 1419-1435.
22. Paice C., (1990), Constructing literature abstracts by computer: techniques and prospects, *Information Processing and Management 26*(1), 171–186.
23. Paice C.D., Jones P.A. (1993), The identification of highly important concepts in highly structured technical papers, in Proceedings of the ACM SIGIR'93, 123-135.
24. Saggion H., Lapalme G., Generating Indicative-Informative Summaries with SumUM. Computational Linguistics. December 2002, 28(4), 497-526.
25. Stairmand M., (1996)., *A Computational Analysis of Lexical Cohesion with Applications in Information Retrieval,* PhD, Department of Language Engineering, UMIST Computational Linguistics Laboratory.
26. Swales J. (1990), Genre Analysis: English in academic and research settings, Cambridge University Press, 1990.
27. Swan R., Allan J., (2000), Automatic generation of overview timelines. Proc. *23rd ACM SIGIR conference*, ACM Press New York, NY, USA, 49–56.
28. Salanger-Meyer F. (1990), Discoursal movements in medical English abstracts and their linguistic exponents: a genre analysis study, INTERFACE: *Journal of Applied Linguistics* 4(2), 1990, 107 – 124.
29. Teufel S., Moens M. (2002), Summarizing scientific articles: Experiments with relevance and rhetorical status, *Computational Linguistics*, 2002, 28(4), 409-445.
30. Teufel S. (1999), *Argumentative Zoning:Information Extraction from Scientic Text*, PhD Dissertation, University of Edinburgh, 352p.

Appendix.

Examples of positive and negative sentences tagged by the automata for sentence classification

Pattern	Pos_example	Neg_example
Results	Model comparisons indicate that the age of the young population of these galaxies does not vary with radius. We find that the slope of composite LFs becomes flatter toward a redder color band. We find that the spectral classification of quasars is redshift and luminosity dependent;	
Conclusions	Hence, we claim the possible universality of the color of the galaxies on the red sequence. Therefore, the existence of the Gunn-Peterson trough by itself does not indicate that the quasar is observed prior to the reionization epoch. We therefore conclude that the point source is likely to be a fifth lensed image of the source quasar.	With this large sample, we have reached the following conclusions. Our analysis leads to the following conclusions**:** Our findings are as follows**.** One method is to search for gaps in the Gunn-Peterson absorption troughs of luminous sources.
Future_work	Further host galaxy observations will be needed to refine the significance of this result. We emphasize the need for further observations of SNe in the	I will review some of the latest developments on cosmological reionization and suggest, in a somewhat more personal way, that the universe may

	rest-frame UV to fully characterize, refine, and improve this method of SN type identification. Future work needed to extend this selection algorithm to larger redshifts, fainter magnitudes, and resolved sources is discussed.	be reionized twice in order to paint... No other planned survey will provide so much photometric information on so many stars. The full SDSS data set will include greater than or similar to 1000 SDSS/RASS clusters.
Newthing	In this paper we report the discovery of a new X-shaped radio galaxy with a partially obscured quasar nucleus. We present evidence for eight new clumps of blue horizontal branch stars discovered in a catalogue of these stars compiled from the Sloan Digital Sky Survey by Sirko et al. and published in 2004. The OLS-lens survey: the discovery of five new galaxy-galaxy strong lenses from the SDSS.	However, only more extensive optical photometry and a detection of its spin or spin-orbit beat frequency can confirm this classification. Detection of quasar clustering anisotropy would confirm the cosmological spacetime curvature that is a fundamental prediction of general relativity. Here we present the New York University Value-Added Galaxy Catalog (NYU-VAGC), a catalog of local galaxies (mostly below z approximate to 0.3) based on a set of publicly released surveys matched to...
Objective	This paper describes spectra of quasar candidates acquired during the commissioning phase of the Low-Resolution Spectrograph of the Hobby-Eberly Telescope. We present results from 1 month, 3 year, and 10 year simulations of such surveys. We investigate the luminosity dependence of quasar clustering, inspired by numerical simulations of galaxy mergers that incorporate black hole growth.	
Related_work	In contrast to past findings, we find that not all M7 - M8 stars are active. Our results are in excellent agreement with recent determinations of these relations by Mandelbaum et al. using galaxy-galaxy weak lensing measurements from the SDSS. Unlike previous work, however, we are able to detect structures in the lens associated with cluster galaxies.	This distribution has been found to have fractal dimension, D, approximately equal to 2.1, in contrast to a homogeneous distribution in which the dimension should approach the value 3 as the scale is increased.
Hypothesis	Knowing that all three methods can have significant biases, a comparison can help to establish their (relative) reliability. A combination of all three effects may better explain the lack of Lyalpha absorption reduction. A larger sample of QSO pairs may be used to diagnose the environment, anisotropy, and lifetime distribution of QSOs. We estimate that the SRN background should be detected (at 1sigma) at Super-K in a total of about 9 years (including the existing 4 years) of data.	Redshifts may have been assigned to some QSOs due to misidentification of observed lines, and unusual spectra should be particularly investigated in this respect. This estimate is based on small-sample statistics and should be treated with appropriate caution. The revision should be taken into account in any future analysis of the source number density of UHECRs based on the ORS.

Frequency & Markup Analysis for Terminological Ontologies

Dr. Roman Schneider, Institute for German Language (IDS), Mannheim/Germany

Abstract

The demonstration's objective is to describe the current activities at the Mannheim Institute for German Language regarding the implementation of a domain-specific ontology for German Grammar. This database-driven ontology was built by analyzing semantic and structural markup in a specialist hypertext corpus. In order to demonstrate the practical use for information retrieval, we outline the semantic retrieval interface to the *grammis* web information system.

Modelling Relationships

Concepts can be connected - permanently, temporarly or situationally - by most different semantic relations. Beisswenger et al. (2004) introduce termsets for the connection of similar terminological concepts. We stick to this idea, but expand the model by adding some theory-related attributes and, secondly, allowing the explicit linking of individual concepts belonging to different termsets. Figure 1 illustrates our model. It contains three termsets, indicated by dotted border lines. The bottom termset contains the two concepts {Verbgruppe} and {Verbalphrase}, recognizable by rectangles with rounded corners. {Verbgruppe} is characterized by a theory-related attribute named "IDS", meaning that it is used primarily when referring to the IDS Grammar of German Language. The concept {Verbalphrase} consists of four lexical entries: 1. {Verbalphrase} with a PT-marker for Preferred Term and with a language attribute (German). 2. {Verbphrase} linked to the former by a synonymy relation. 3. {VP} linked by a abbreviation relation. 4. {Verb Phrase} with a language attribute (English) and linked with a translation relation. The complete termset, which additionally may be characterized by an optional and inheritable attribute for the grouping of co-hyponyms, is linked with its hyperonym termset by a BTG (Broader Term Partial) relation.

In order to clarify the benefit of linking not only termsets, but also individual concepts, our example illustrates the relationships between {Phrase} (engl. "phrase") and {Satz} (engl. "sentence"). Basically, the corresponding termsets are connected with the help of a Broader Term Partitive (BTP) relation (meronymy). Beyond this, since generative grammars usually classify sentences (complementizer phrases) as phrases, only these two concepts - singled out by a theory-related attribute - are linked by an Narrower Term Generic (NTG) relation (hyponymy). This fact, explicitly coded within the ontology base, should facilitate communication between people or computer systems using different terminological vocabularies. Furthermore, we use standard relationship types like Related Term (RT) for the linking of termsets that are associated in some way, but without the necessity of deeper relationship explanation. Good examples are {Wortschatz} (engl. "vocabulary") and {Wortschatzerweiterung} (engl. "vocabulary extension") or {Fokus} (engl. "focus") and {Fokuspartikel} (engl. "focusing adjunct"): Focusing adjuncts mark the focus. Because we do not see a need for introducing a special relationship type for this, we simply call them RTs.

Detecting Concepts

Concept selection is probably one of the most challenging subtasks within the ontology lifecycle, and can be done using three different approaches: 1. Intellectual/manual compilation of all relevant domain concepts by human experts. 2. Use of statistical methods on a given representative corpus. 3. Use of linguistic methods. Usually, the selection depends primarily on project-specific factors, preferences, and objectives. Recourse to human knowledge demands a relatively large amount of time, but generally guarantees high quality. Statistical methods depend on sufficiently large corpora as well as on long-time experience in fine tuning algorithms and parameters. Linguistic methods, e. g., the use of morpho-syntactic information, succeed only if parser, tagger, and lexicon supply reliable results.

For the detection of concept, we successfully used a combined method comprising statistical exploration, linguistic analysis as well as manual post-editing. The underlying specialist language corpus was made up of XML-structured hypertexts from the *grammis* and *ProGr@mm* information systems hosted at IDS. Altogether we included a total of about 2,000 hypertext nodes with almost 1,000,000 wordforms (N_{SL}). Furthermore, we used COSMAS (Corpus Search,Management and Analysis System, http://www.ids-mannheim.de/cosmas2/) for exploring 160 general language corpora with more than 1.6 billion wordforms (N_{GL}). In the following, we present our six steps for concept acquisition:

1. Frequency analysis of specialist language corpus: The specialist language (SL) hypertexts are used as input. We tokenize the corpus and collect frequency information for each token (f_{SL}). Stop words are omitted. Output is an ordered list with two columns (wordform, f_{SL}).
2. Markup analysis: We use the output list from step 1 as well as XML-coded meta information from the specialist corpus as input. Wordforms appearing in the most prominent hypertext structures - i. e., in titles, subtitles, definitions, and semantically typed hyperlinks - receive a ranking bonus. Output is an accordingly modified f_{SL} list.
3. Frequency analysis of general language corpus: We use the output list from step 2 together with the COSMAS-maintained general language (GL) corpora as input. For each wordform, we calculate the GL-frequency value (f_{GL}). Output is a list with three columns (wordform, modied f_{SL}, f_{GL}).
4. Weirdness value: We use the ouput list from step 3 as input and compute a "weirdness" value $\tau(w)$ for each wordform (see Gillam/Tariq/Ahmad 2005). The computed value tells us which wordforms appear significantly more frequent in the specialist corpus than in the general language corpus. Higher values indicate interesting wordforms, i. e., concept candidates.

$$\tau(w) = \frac{N_{GL} f_{SL}}{f_{GL} N_{SL}}$$

5. Collocation analysis: We use the list from step 4 as well as the SL-corpus as input. We examine the co-occurrence of concept candidates by using varying environments (sentences, paragraphs, hypertext nodes). Even basic vectors can be detected: given that concept candidate X appears more frequent in conjunction with concept candidate Y than Y together with X, then we may say that Y stands for a more general concept than X. Output is a set of concept candidate clusters, i. e., collocations of concept candidates.
6. Relationship assignment: Input is the cluster set from step 5. Now a human expert has to decide which concept candidates should be considered as domain-specific and which relations should be coded on the basis of our cluster set. Output is a tentative terminological net, which already contains some partial hierarchies.

Database Implementation and Retrieval

When it comes to database implementation, the number of possible modelling strategies, methods, and systems is enormous. Assuming that reliable and high-performance solutions require professional database management systems (DBMS), we decided to adopt the object-relational DBMS already in use for the *grammis* web information system. For portability reasons, we designed our conceptual data model according to the well-established entity-relationship paradigm, and used the relational approach for database implementation. Figure 2 shows our model. The further implementation process is quite straightforward.

Obviously, a major benefit of using integrated ontologies is their support for text classification and retrieval. Traditional full text search, based on the vector model, is limited in terms of semantic markers. Most users find it difficult to formulate queries which are well designed for retrieval purposes. Nevertheless, users of complex information systems often consider full text search as the preferred access option. But it supplies satisfying results only if humans and computer speak the same language, i. e., share a common terminology. For our system this means: if the user types in "Ergänzung", the system should realize that this is synonym for "Komplement" (engl. "complement"), and it should link it to "Valenz" (engl. "valency"). The query is expanded, and the result set increases. In order to avoid a disproportional increase, on a certain level the reverse strategy of query reformulation seems necessary: if the system recognizes that a search term ranks high in the ontological hierarchy, e. g., "Valenz", it should offer a set of subordinated terms, e. g., "Verbvalenz", with probably less retrieved documents.

A graphical representation of the ontology structure assists the ontology author through all phases of the ontology lifecycle. Besides, it helps end users in situations when they cannot precisely formulate their information need or just want to browse the whole system. For these reasons, we included a graphical retrieval and navigation frontend. Figure 3 illustrates the functionality: in the center we see the currently accessed termset. Above, bordered by specifically colored block elements and serving as hyperlink anchors, the immediately superordinated hyperonymes and holonymes can be found; below are hyponyms and

meronymes. Associated concepts are displayed also. By pointing and clicking, users activate the different relations and change their position within the informational space.

Since our database-driven ontology is directly connected to the whole *grammis* information system, the frontend comprises appropriate retrieval options, mapping user input to standard SQL statements. By drag-and-drop, users are allowed to insert any term from the graphical structure into one of the three containers on the right side. The system then sifts through the hypertext base as well as through the bibliography and all dictionaries. The number of hits is immediately displayed next to the container; the actual result set is presented by request in a separate pop-up window. Results of combined queries are shown between the containers.

Conclusion

Our approach already allows for the integration of different terminological systems and languages, and thereby supports international scientific collaboration and research. We believe that multilingual, theory-spanning domain ontologies will be a clear asset for all projects related to the vision of the semantic web. Our aim is not so much the formal unification of ontological models, but rather the accurate representation of domain-specific concepts and relationships with respect to our retrieval and classification goals. We accept that there is not self-evident way of dividing the world - or even small parts of it – into concepts. Especially in terminology we often deal with hardly dissolvable antagonisms. Nevertheless, our ontology's hierarchical backbone should be integrable with almost any upper ontology, and convertible to most terminology exchange formats and Terminology Management Systems.

Figures

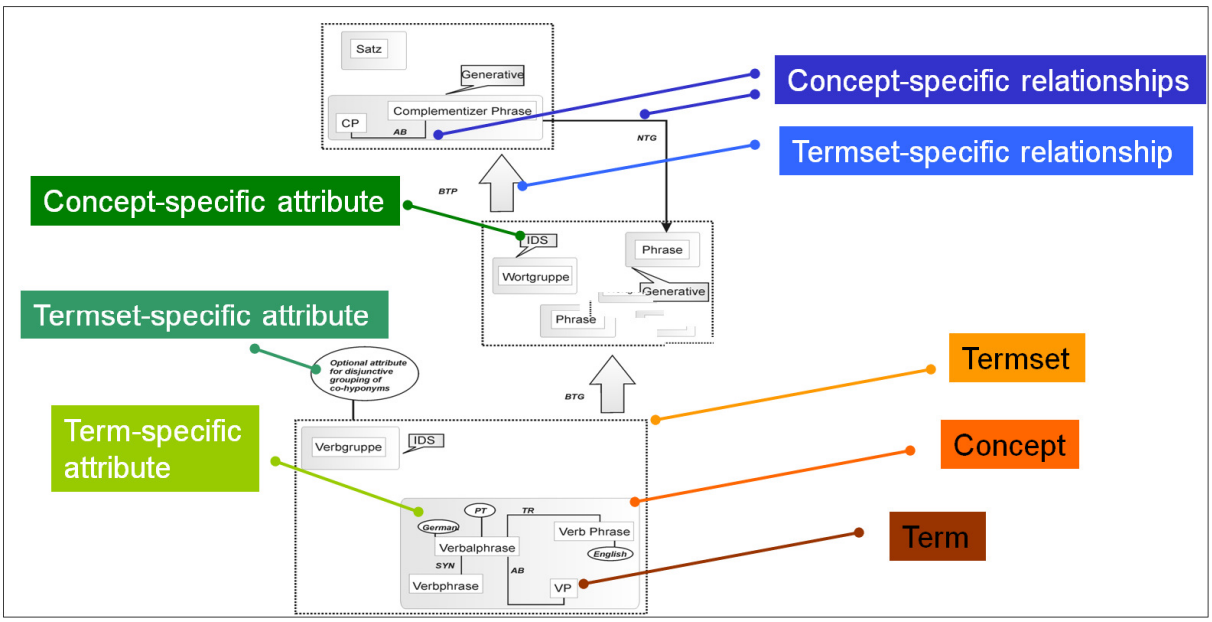

Figure 1: Relationship modelling for concepts and termsets

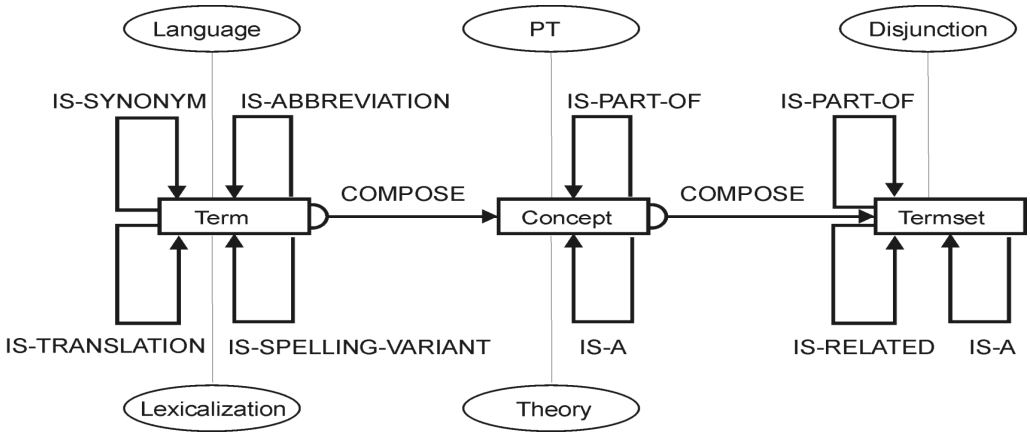

Figure 2: ER data model

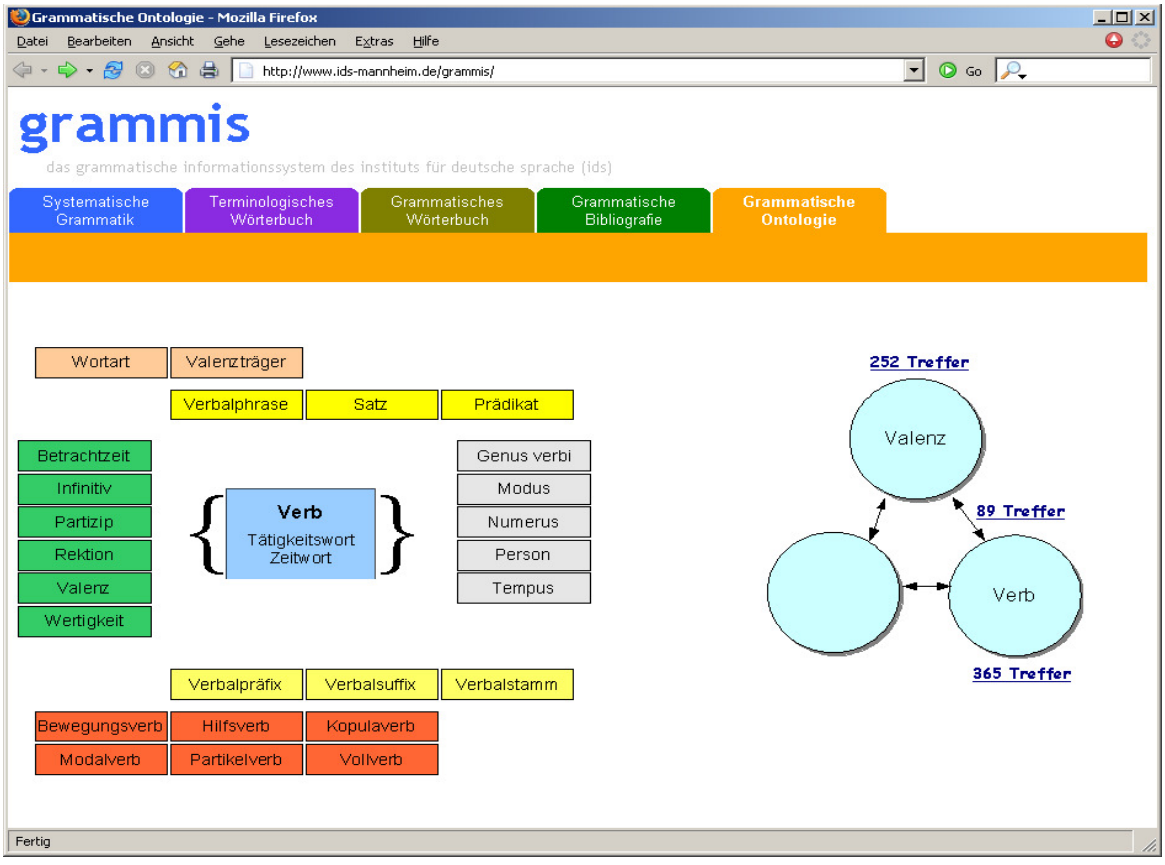

Figure 3: Retrieval frontend; http://hypermedia.ids-mannheim.de/pls/public/ontologie.html

Bibliography

Beisswenger, M., Storrer, A., and Runte, M. (2004). Modellierung eines Terminologienetzes für das automatische Linking auf der Grundlage von Wordnet. LDV-Forum, 19(1/2):113–125.

Gillam, L. / Tariq, M. / Ahmad, K. (2005): Terminology and the construction of ontology. Terminology, 11(1). 55–81

Schneider, R. (2007): A Database-driven Ontology for German Grammar. In: Rehm, G. / Witt, A. / Lemnitzer, L. (Ed.): Data Structures for Linguistic Resources and Applications. Proceedings of the Biennial GLDV Conference 2007, Tübingen: Narr. S. 305-314.