Exploiting Ontological Relations for Automatic Semantic Tag Recommendation

Panos Alexopoulos
IMC Technologies S.A.
360A Kifissias Str. & 2
Marathonos Str. , 15233,
Athens, Greece
palexopoulos@imc.com.gr

John Pavlopoulos
Department of Informatics,
Athens University of
Economics and Business
Kodrigktonos 12, Athens,
Greece
annis.pavlo@gmail.com

Manolis Wallace
Department of Computer
Science and Technology,
University of Peloponnese
End of Karaiskaki St., 22100,
Tripolis, Greece
wallace@uop.gr

Konstantinos Kafentzis
IMC Technologies S.A.
360A Kifissias Str. & 2 Marathonos Str. , 15233, Athens, Greece
kkafentzis@imc.com.gr

ABSTRACT

In this paper we propose a novel method for automatically generating and recommending semantic tags for text documents, namely terms that reflect the intended meaning of the document in an accurate and complete way. Our approach is based on the utilization of existing domain knowledge, in the form of ontologies, and particularly in the selection and exploitation of those ontological relations that are most appropriate for the given tagging scenario and domain. Experimental evaluation of the method with significant number of documents and high volume of ontological knowledge shows a high level of accuracy as far as tag identification is concerned.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing

General Terms

Algorithms, Experimentation

Keywords

Semantic Tagging, Ontologies

1. INTRODUCTION

Tagging is a textual annotation technique that involves assigning to a document terms and phrases that are representative of its semantic content. The term "representative" may have a different interpretation depending on the reason why tagging is employed. As suggested in [11], tagging is typically used with the explicit intent of:

- Identifying the concepts to which various terms and phrases of the document belong.
- Classifying a document by means of concepts that represent meaningful categories for the document user (e.g. determining whether a document is about sport events or films).
- Characterizing a document by means of proper adjectives that denote some kind of judgment(e.g. "positive", "negative").
- Summarizing a document's content by means of keywords that constitute a representative description of what the document is specifically about (e.g. determining for which sport event or for which film is a document about).

Regardless of its intent, tagging has been traditionally approached using systems for vocabulary control (indexing languages, thesauri and/or ontologies) and classification systems (e.g. faceted classification systems or taxonomies). These approaches require dedicated human experts who need to read, understand and reflect on the document's content and then decide which tags should be assigned to it. As such, they do not easily scale to the large collections of documents, both in terms of cost, time, and expertise of the human personnel required.

An alternative approach, useful for the Web in which there is too much content for a single authority to classify, is collaborative tagging [4] where users freely determine suitable labels for their resources without relying on any predetermined vocabulary or hierarchy (e.g. Del.icio.us, Bibsonomy, Flickr). Limitations of such systems include, among others, ambiguity (i.e. many tags have multiple meanings) and independence of terms (i.e. each tag is independent of the others and thus inference is not possible).

For the above reasons the development of methods that may generate tags in an automated fashion is highly desirable. With that in mind, we propose in this paper a novel

I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria

framework for automatically generating and recommending to users tags for text documents through the exploitation of domain ontologies. In our approach, we consider tagging not so much as a classification process but rather as a summarization one. This means, for example, that we do not wish to determine whether a given document is about sports or politics but **which specific sport events or politicians it is about**. The challenge in this kind of identification is to be able to distinguish between the keywords that play a central role to the documents's meaning and those that are just complementary to it. For example, a piece of news might make reference to many politicians even when its primary subject is only one of them.

To achieve such a distinction, we follow a "detective-like" approach and we try to find evidence within the document that point towards the correct tag(s). In doing that we assume that we have available for the documents we wish to tag some ontology that describes their domain (e.g. a cinema ontology for film-related documents). Since we wish to tag the documents with specific entities (e.g. films, director, actors etc) we consider as candidate tags the instances of the ontology's concepts and we work on the following two premises:

1. That an instance is more likely to represent the text's meaning when there are many ontologically related to it instances in the text. For example, in a historical document, a given event is a likely tag when the text also contains persons, locations and other events related to this event. Similarly, in a document about cinema, a given film is a likely tag when the text also contains persons involved in the film (directors, actors, characters etc.).

2. That not all relations in a domain ontology are equally important in the above process. For example, the relation *hasCharacter* between a film and a film character is more important than the relation *hasDirector* between a film and a director when it comes to determining whether the text actually refers to a given film. The reason is that the reference within a specific film text of a film character that wasn't part of it, is a rather rare fact.

Our proposed framework builds upon these two premises by defining two components:

- A **tagging context model** that models the relative importance of ontological relations to the tag identification process.

- A **tag recommendation process** that determines, for a given text, the ontology instances that are potential tags for it along with a **confidence score** for each of them. This score is used to rank the identified tags and recommend to the users those that have the highest confidence.

The rest of the paper is organized as follows. In section 2 we present related work while in section 3 we present in a detailed manner the components of our proposed tagging framework including the tagging context model and the tag identification process. In section 4 we provide an indicative use case of our method in the film domain and in section 5 we present and discuss experimental results regarding the evaluation of our method's effectiveness in recommending tags. Finally, in section 6 we list our concluding remarks and outline potential future work.

2. RELATED WORK

Our proposed tagging framework generates tag recommendations based on domain ontologies. In literature there are many examples of tag recommender systems [7] [14] [3] [5] [10], but only few of them use ontologies.

A work in which ontologies are used for tagging is that of [15] where the authors present ePaper, a system that uses a hierarchical news ontology as a common language for content based filtering in order to classify news items and to deliver personalized newspaper services on a mobile reading device. In another work [11] the authors propose a tag recommendation process based on keyphrase extraction and ontology reasoning. In particular, their approach involves the utilization of linguistic and statistical processing for determining keyphrases that could be potential tags and the exploitation of domain ontologies for suggesting tags that are not present within the document. For the latter, they use a reasoning mechanism based on the subsumption relationship between concepts (is-a) and the spreading activation algorithm of [12].

A similar approach is presented in [9] where the authors discuss ontology-based document annotation for the purpose of semantic indexing and retrieval. The method they propose expands, both syntactically and semantically, concept descriptions taken from the domain ontology in order to enhance matching in the retrieval process. The syntactic expansion is based on lexical resources (e.g. Wordnet) while the semantic one on a concept exploration algorithm that is applied on the ontology.

In [2] the authors propose GoNTogle, a framework for document annotation and retrieval, built on top of Semantic Web and Information Retrieval technologies. For the annotation part, GoNTogle supports the automatic annotation of a whole document or parts of it with ontology concepts through a learning method based on *weighted kNN* classification [8] that exploits user annotation history and textual information to automatically suggest annotations for new documents.

In [1] the authors suggest an approach to generate semantic tag recommendations for documents based on Semantic Web ontologies and Web 2.0 services. In particular, their proposed process starts with the extraction of document entities through the utilization of Web 2.0 services (such as Yahoo's Term Extraction service and their transformation into a topic map using SKOS vocabulary (Simple Knowledge Organisation System) [6]. Then, the topics of this topic map are matched, based on document classification methods, to instances of some domain ontology expressed according to the PIMO ontology [13]. The matching pairs are shown to the users as tag recommendations and they decide whether to accept or reject them.

With regard to the above, our work is differentiated either in its focus, its approach or both. More specifically, in contrast to [11], [9] and [2], our goal is not to detect which ontology concepts are related to the document (either directly through the existence of their instances within the text or indirectly through ontological relations) but which ontology instances best describe the subject of the document. Thus, for example, whereas the method of [11] would recommend as tags for a software document concepts like "Programming Languages" or "Software Notations" in an effort to describe its topic (that is the use case described in [11]), our method would try to determine for which specific language or notation is the document actually about.

Furthermore, our method differs from all the above approaches in the way it exploits the domain ontology for identifying appropriate tags. As it will be shown in the next sections, any relation defined within the ontology may be exploited, not just the subsumption one. Also, it does that in a controlled and scenario-specific way so that only those relations that are useful for the tagging task in hand are actually utilized.

3. TAG RECOMMENDATION FRAMEWORK

3.1 Problem Setting and Tagging Process

For the purposes of this paper we define an ontology as a tuple $O = \{C, R, dom_R, range_R, I, i_C, i_R\}$ where

- C is a set of concepts
- I is a set of instances
- R is a set of binary relations that may link pairs of concept instances.
- i_C is a concept instantiation function $C \rightarrow I$
- i_R is a relation instantiation function $R \rightarrow I \times I$
- dom_R is a function $R \rightarrow C$ that returns for a given relation $r \in R$ the concepts which the potential subjects of the relation may instantiate.
- $range_R$ is a function $R \rightarrow C$ that returns for a given relation $r \in R$ the concepts which the potential objects of the relation may instantiate.

Given a text document, we wish to determine a tag confidence function $f : I \rightarrow [0, 1]$ which returns for a given ontological instance the confidence that it is a suitable tag for the document. The process we propose for this comprises the following steps:

1. We select the ontology concepts whose instances are going to be used as candidate tags (e.g. films, film directors, events etc.). The selection depends on the domain and on the particular requirements of the application scenario.
2. For each of these concepts we determine the ontological relations that have this concept within their domain and we define their relative importance in the tagging process by means of a **Tagging Context**.
3. We extract from the text the terms that match to instances of the tag concepts as well those that match to instances of the concepts that fall within the range of the tag ontological relations.
4. Using the ontological relations we derive from the above set of terms those that are related to instances of the tag concepts. The derived terms comprise the **candidate tags** for the text.
5. For each of the candidate tags we calculate its **Ontological Support**, namely the degree to which the instances found within the text imply that the candidate tag actually characterizes the text.
6. For each of the candidate tags we calculate its **Ontological Ambiguity**, namely the degree to which the ontology instances that are found within the text and support the tag, support other tags as well.
7. Using the two scores, we derive for each candidate tag a **Tag Confidence** function that indicates the confidence that the tag actually characterizes the text.

In the following paragraphs we elaborate on each of the above steps.

3.2 Tagging Context

The **Tagging Context** defines for each ontology concept which relations and to what extent should be used for determining the degree to which the concepts's instances are supported by other ontology instances. This is important as, as mentioned in the introduction, not all relations contribute necessarily the same to the identification of the text's tags.

More formally, given a domain ontology O, a tagging context is defined as a function $g : C \times R \rightarrow [0, 1]$. If $c \in C$, $r \in R$ and $c \in dom_R(r)$ then $g(c, r)$ is the degree to which the existence, within the text, of an instance that instantiates some concept $c_i \in range_R(r)$ should be considered an indication that the related to it through r instances are candidate tags for the text.

For example, if $C = \{Film, Actor\}$, $R = \{hasActor\}$, $dom_R(hasActor) = \{Film\}$ and $range_R(hasActor) = \{Actor\}$ then $g(Film, hasActor)$ denotes the degree to which the existence of a specific actor within a text should be considered an indication that the films he/she has played in are appropriate tags for the text.

Obviously the importance degree of a relation is not to be determined on its own but in relevance to the degrees of the other ontology relations. In any case, the intuitive definition of a tagging context for the domain and application scenario in hand is an important factor for the success of our proposed method.

3.3 Candidate Tag Extraction

In this step of the method we seek to determine the ontology instances that are potential tags for the text. For that we consider the following sets:

- The set $C_{tag} \subseteq C$ which contains the concepts whose instances are going to be used as tags.
- The set $I_{tag} \subseteq I$ which contains the instances that instantiate the concepts of C_{tag}.
- The set $I_{support} \subseteq I$ which contains the instances that are related to instances of I_{tag} through a set of relations $\{r_1, r_2, ..., r_n\}$ for which $g(c, r_i) > 0, c \in C_{tag}$.
- The set of instances $I_{text} \subseteq (I_{tag} \cup I_{support})$ that contains those instances of $I_{tag} \cup I_{support}$ that are found within the text.

Then, for each $i \in I_{text}$ we derive all the instances from I_{tag} that are related to it through the relations $\{r_1, r_2, ..., r_n\}, r_i \in$

R for which $g(c, r_i) > 0, c \in C_{tag}$. The result is set of candidate tags $I_{candidate} \subseteq I_{tag}$ each of which is accompanied by a function $s : I_{candidate} \times I_{text} \rightarrow [0, 1]$ that returns for each pair of a candidate tag and a text extracted instance the degree to which the latter "supports" (i.e. indicates) the former. If $j \in I_{candidate}$, $i \in I_{text}$, and i is related to j through the relations $\{r_1, r_2, ..., r_n\}$ then $s(j, i) = max(g(c, r_i))$, $i \in i_C(c)$. In other words, from all the relations through which i supports j we consider the one with the maximum importance as defined in the tagging context.

3.4 Tag Ontological Support

The ontological support of a candidate tag is the degree to which the instances found within the text imply that the candidate tag actually characterizes the text. Intuitively, this degree is analogous to the number and support degrees of the tag's supporting text instances (as derived from function $s : I_{candidate} \times I_{text} \rightarrow [0, 1]$). Thus, given a candidate tag $j \in I_{candidate}$ its ontological support can be calculated as follows:

$$S(j) = \frac{\sum_{i \in I_{text}} K(j, i)}{\sum_{m \in I_{candidate}} \sum_{i \in I_{text}} K(m, i)} * \sum_{i \in I_{text}} s(j, i) \quad (1)$$

where $K(m, i) = 1$ if $s(m, i) > 0$ and 0 otherwise.

In other words, the overall support score for a given candidate tag is equal to the sum of the tag's partial supports (i.e. function s) weighted by the relative number of the instances that support it. This weighting is important as our aim is to compute for the tag a support that is relative to the supports of the other tags.

3.5 Tag Ontological Ambiguity

The ontological ambiguity of a tag is the degree to which the ontology instances that are found within the text and support the tag, support other tags as well. For a given tag, the more additional tags its support function (i.e. function s) also supports, the higher is the ambiguity of the tag.

The ambiguity between two tags $j_1, j_2 \in I_{candidate}$ is calculated as follows: First we find the set of text extracted instances that support j_1, namely $I_{j_1} = \{t_1, t_2, ..., t_n\}, t_i \in I_{text}$ for which $s(j_1, t_i) > 0$ and in a similar fashion the set I_{j_2} that supports j_2. Then we consider the set that commonly supports the two tags, namely $I_{j_1,j_2} = I_{j_1} \cap I_{j_2}$. After that we calculate the contribution degree of this set to each of the wholes through the formulas:

$$b(I_{j_1,j_2}, I_{j_1}) = \frac{\sum_{t_i \in I_{j_1,j_2}} s(j_1, t_i)}{\sum_{t_i \in I_{j_1}} s(j_1, t_i)} * e^{\frac{|I_{j_1,j_2}|}{|I_{j_1}|}} \quad (2)$$

$$b(I_{j_1,j_2}, I_{j_2}) = \frac{\sum_{t_i \in I_{j_1,j_2}} s(j_2, t_i)}{\sum_{t_i \in I_{j_2}} s(j_2, t_i)} * e^{\frac{|I_{j_1,j_2}|}{|I_{j_2}|}} \quad (3)$$

Given these, the ambiguity of the pair is given by the formula:

$$a(j_1, j_2) = \frac{b_1 + b_2}{2} * e^{\frac{-|b_1 - b_2|}{\sigma}} \quad (4)$$

where $b_1 = b(I_{j_1,j_2}, I_{j_1})$ and $b_2 = b(I_{j_1,j_2}, I_{j_2})$ and σ is a tuning variable. The intuition behind this formula is the following: If the two contributions are high and comparable then the ambiguity is high. If they are low and comparable then the ambiguity is low and, finally, if they are non-comparable (i.e. one relatively high and one relatively low) then again the ambiguity is low. Figure 1 illustrates this behaviour.

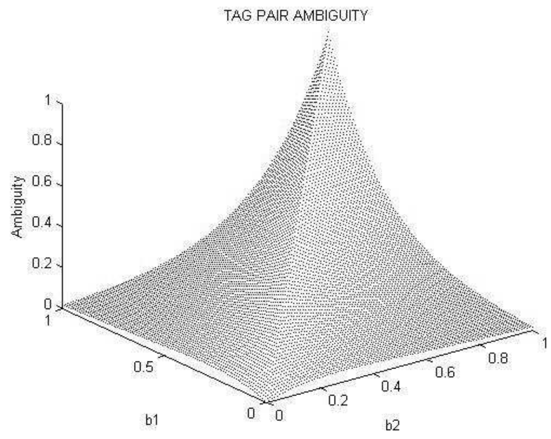

Figure 1: Tag Pair Ambiguity

Having found the ambiguities of the pairs, we then calculate the overall ambiguity of a candidate tag $t \in I_{candidate}$ as follows:

$$A(t) = \frac{\sum_{t_i \in I_{candidate}} a(t, t_i)}{\sum_{t_i \in I_{candidate}} \sum_{t_j \in I_{candidate}} a(t_i, t_j)} \quad (5)$$

3.6 Tag Confidence

In order to derive an overall score about a given tag's likelihood that it actually characterizes the text we use its ambiguity score in order to "adjust" its initial ontological support. The intuition here is that the more ambiguous is the tag, the less "reliable" is its support. Thus, the formulas we use for calculating the overall tag confidence are the following:

$$f(t) = S(t) * e^{w*(1-A(t))} \quad (6)$$

$$f_{norm}(t) = \frac{f(t)}{\sum_{t_i \in I_{candidate}} f(t_i)} \quad (7)$$

where $w \in [0, 1]$ is a weight that adjusts the influence of the ambiguity score to the total confidence and $f_{norm}(t)$ is the normalized tag confidence. The latter, which is not an absolute measure of tag confidence but rather a relative one, enables the ranking of the candidate tags and the recommendation to the users of those that have the the highest confidence.

4. USE CASE SCENARIO

To illustrate the applicability of our proposed method we consider a use case scenario in which we have a document that contains a film review and we wish to automatically identify for which film it is about. The review's text (which is taken from IMDB and is about the film **"Steel"**) is as follows:

*"How's this for diminishing returns? In BATMAN AND ROBIN, George Clooney battled Arnold Schwarzenegger. In SPAWN, it was Michael Jai White versus John Leguizamo. In STEEL, the third and presumably *final* superhero stretch of the summer, Shaquille O'Neal dons a high-tech, hand-crafted suit of armor to combat the earth-shaking, world-shattering, super-duper-ultra evil menace of... Judd Nelson? Holy economical casting! The latest synergistic teaming of Warner Brothers and D.C. Comics is a strictly third-rate affair, with a forgettable hero (Shaq, all size and smiles in his ridiculous costume) performing forgettable feats (stopping a mugger, shooting back at bad guys) while in pursuit of a forgettable villain (Nelson's revenge-seeking arms dealer, whose sole super-power appears to be the ability to stifle laughter among those actors sharing scenes with him). Pure cheese, but it's well-intended. There's lots of family love and a majority of minority characters and one disabled person (Annabeth Gish) that the story makes a bit of an embarrassing big deal about, but later redeems itself by giving her a kick-ass laser-firing wheelchair. (Let's see *her* action figure go up against Share-a-Smile Barbie.) The amount of gratuitous violence is quite reasonable and there's enough scattered laughs to hold a bored parent's attention. (I enjoyed Richard Roundtree's SHAFT reference, a couple of hilarious comments about the Internet, and one whopper of an Ahnuld gag at the end.) If only the lack of realism wasn't *quite* as distracting. I know, I know, not exactly a valid complaint in a movie based on a comic book, but, good God, a Man of Steel with an exposed mouth and lower jaw? And who spends half of his time engaged in gunfights? Sigh. Written and directed by Kenneth Johnson, an old TV hack whose credits include "Alien Nation" and "The Incredible Hulk." Eat the hot dog, don't be one"*

In order to apply our method we consider a film ontology comprising the concepts C = {*Film, Director, Actor, Character*} and the relations R = {*hasDirector, hasActor, hasCharacter*}. Instances of the ontology are taken from the film ontology from Freebase[1]. Then we define the following tagging context:

- $g(Film, hasDirector) = 0.5$
- $g(Film, hasActor) = 0.4$
- $g(Film, hasCharacter) = 0.8$

In other words, we consider the *hasCharacter* relation as the most indicative for the identification of a film, followed by the *hasDirector*. Given this context we then proceed with step 3 of paragraph 3.1, namely we extract from the text instances of the concept Film as well instances of the concepts Director, Actor and Character since they fall within the range of the tagging context's relations. In practice, these extracted instances are going to be the "evidence" on which the identification of the correct tag is going to be based.

Using these instances and the relations of the tagging context we apply the procedure of paragraph 3.3 and we determine the candidate tags along with their supporting ontology instances. Examples of such tags include the film "Batman and Robin" (supported, among others, by the actors Clooney and Schwarzenegger) and the film "Steel" (supported, among others, by the director Kenneth Johnson and the actor Shaquille O'Neal).

[1]http://www.freebase.com/schema/film

Table 1: Method Results for Use Case Example

Tag	Support	Ambiguity	Confidence
Steel	**0.064**	**0**	**0.084**
Batman and Robin	0.068	0.35	0.077
The Villain	0.064	0.38	0.071
Revenge	0.064	0.43	0.069
Spawn	0.064	0.43	0.069
Brothers	0.047	0.11	0.058
Batman	0.047	1	0.04

Subsequently, we apply the formulas of paragraphs 3.4, 3.5 and 3.6 in order to determine for each candidate tag its ontological support, its ambiguity and its overall confidence respectively. Table 1 shows the results of this process for the candidate tags with the highest scores.

From these results one can easily see that the proposed method manages to recommend the correct tag "Steel" with the highest confidence. This is achieved not only because, compared to most other tags, "Steel" has greater ontological support (meaning that there are more "indicative" entities of it within the text) but, most importantly, because its ambiguity, compared to that of "Batman and Robin" is greater, thus making the latter less reliable. That is the reason why, despite the fact that the tag "Batman and Robin" has greater support, "Steel" earns higher confidence in the end.

5. FRAMEWORK EVALUATION

In order to provide a more comprehensive evaluation of our method we performed an experiment involving the tagging of multiple documents containing film reviews. Our aim, as in the example above, was to identify, through our method, the film each review was about. For that we used 1000 reviews, randomly selected from a set of 25000 IMDB reviews[2], as well as the film ontology from Freebase described in the previous section. The ontology comprised about 148000 films, 145000 actors, 63000 characters and the relations between them (film with directors, films with actors and films with characters).

For each document we generated two set of recommended tags: One using a basic keyphrase extraction methodology (in which we assumed that the most frequent film within the text is the one the text is talking about) and one using our proposed method. In the latter we used the tagging context of the previous paragraph and a value 0.5 for the ambiguity weight. We then measured the success of each method by measuring the number of cases in which the tag with the highest score was the correct one.

As table 2 shows, our method outperformed significantly the baseline one (i.e. the one based on term frequency) and achieved a very high success rate. This rate validates our initial hypothesis that the consideration of ontological knowledge (and especially ontological relations) may facilitate more effective generation of relevant semantic tag recommendations for documents.

6. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a novel method for automatically generating and recommending semantic tags for text

[2]http://www.cs.cornell.edu/people/pabo/movie-review-data

Table 2: Tagging Method Evaluation Results

Approach	Success Rate
Baseline	45%
Proposed	89%

documents in an effort to summarize the intended meaning of their content. Our approach has been based on the customized utilization of domain-specific ontological relations for extracting and evaluating "evidence" from within the text that may identify the correct tag(s) in the given tagging scenario. A comprehensive experimental evaluation of the method highlighted its high effectiveness for the tag recommendation task.

Future work will focus on further experiments where we will seek to establish the effectiveness of our method in more complex and ambiguous domains and with larger datasets. Also, since the performance of our proposed method is in direct relation with the amount and quality of information within the domain ontology, an important task is to see how our method should be adapted in order to cope with knowledge incompleteness and/or inconsistency.

7. REFERENCES

[1] B. Adrian, L. Sauermann, and T. Roth-Berghofer. Contag: A semantic tag recommendation system. In T. Pellegrini and S. Schaffert, editors, *Proceedings of I-Semantics' 07*, pages pp. 297–304. JUCS, 2007.

[2] N. Bikakis, G. Giannopoulos, T. Dalamagas, and T. Sellis. Integrating keywords and semantics on document annotation and search. In *Proceedings of the 2010 international conference on On the move to meaningful internet systems: Part II*, OTM'10, pages 921–938, Berlin, Heidelberg, 2010. Springer-Verlag.

[3] J. Gemmell, T. Schimoler, B. Mobasher, and R. Burke. Hybrid tag recommendation for social annotation systems. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 829–838, New York, NY, USA, 2010. ACM.

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

[5] K.-n. Hassanali and V. Hatzivassiloglou. Automatic detection of tags for political blogs. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, WSA '10, pages 21–22, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[6] A. Miles and D. Brickley. Skos core vocabulary specification. W3c working draft, World Wide Web Consortium, 2005.

[7] G. Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM Press. paper presented at the poster track.

[8] T. M. Mitchell. *Machine learning*. McGraw Hill, New York, 1997.

[9] S. Nešić, F. Crestani, M. Jazayeri, and D. Gašević. Concept-based semantic annotation, indexing and retrieval of office-like document units. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, RIAO '10, pages 134–135, Paris, France, 2010.

[10] B. Oliveira, P. Calado, and H. S. Pinto. Automatic tag suggestion based on resource contents. In *Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns*, EKAW '08, pages 255–264, Berlin, Heidelberg, 2008. Springer-Verlag.

[11] N. Pudota, A. Dattolo, A. Baruzzo, F. Ferrara, and C. Tasso. Automatic keyphrase extraction and ontology mining for content-based tag recommendation. *International Journal of Intelligent Systems*, 25(12):1158–1186, 2010.

[12] M. R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, 1968.

[13] L. Sauermann. Pimo-a pim ontology for the semantic desktop (draft). Draft, DFKI, 2006.

[14] M. Tatu, M. Srikanth, and T. D'Silva. Tag recommendations using bookmark content. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 96–107, 2008.

[15] L. Tenenbaum, B. Shapira, and P. Shoval. Ontology-based classification of news in an electronic newspaper. In *Proceedings of INFOS 2008, Varna, Bulgaria*, pages 89–97, 2008.