TasTicWiki: A Semantic Wiki with Content Recommendation

Manuela Ruiz-Montiel, Joaquín J. Molina-Castro, and José F. Aldana-Montes

Departamento de Lenguajes y Ciencias de la Computación,
University of Málaga, España
manuela.ruiz.montiel@gmail.com,{jmolina,jfam}@lcc.uma.es

Abstract. Wikis are a great tool inside the Social Web, as they provide the chance of creating collaborative knowledge in a quick way. Semantic wikis are becoming popular as Web technologies evolve: ontologies and semantic markup on the Web allow the generation of machine-readable information. Semantic wikis are often seen as small semantic webs as they provide support for enhanced navigation and searching of their contents, just what the standards of the Semantic Web aim to offer. Moreover, the great amount of information normally present inside wikis, or any web page, creates the necessity of some kind of filtering or personalized recommendation in order to lighten the search of interesting items. We have developed TasTicWiki, a novel semantic wiki engine which takes advantage of semantic information in order, not only to enhance navigation and searching, but also to provide recommendation services.

Key words: semantic wikis, recommender systems, ontologies

1 Introduction

A wiki is a web site with collaboratively edited pages. Users of the wiki perform these editions through the browser, in a quick way and without restrictions. Each page or article has an unique identifier, so they can be referenced from anywhere inside or outside the wiki. The general features of wikis are the following [1]: editing via browser with a simplified syntax -rather than HTML tags-, collaborative editing, non-lineal navigation thanks to a large number of hypertext links to other wiki pages, search functions and support for uploading non-textual contents.

We have developed a wiki engine, TasTicWiki, which seizes semantic technologies in order to offer sophisticated functionalities as well as semantic-enhanced recommendation services, in order to enlighten the tedious searching tasks derived from the potential existence of a vast amount of articles. In the next sections we will explain how this objectives are achieved as well as the architecture and features of TasTicWiki.

2 Semantic Wikis

Semantic Wikis are traditional wikis extended with semantic technologies like OWL or RDF. The goal of this enrichment is to make the available information machine-readable, so presentation, navigation, searching and even edition can be improved in a sophisticated way. This is usually done by adding meaning to the strong linking present in every wiki: the links are not mere hypertext anymore, as they represent meaningful relations among articles, or between articles and data types.

Common features of all approaches to semantic wikis are the following [3]: typing/annotating of links, context-aware presentation, enhanced navigation, semantic search and reasoning support. Some of the existing semantic wikis delegate the responsibility of creating the knowledge base to the final users of the wiki, allowing them to define meaningful relations practically without restrictions. Others rely on already defined ontologies that form the knowledge base, so the relations to be used are defined and restricted from the beginning. In `http://semanticweb.org/wiki/Semantic_Wiki_State_Of_The_Art#Active` we can find a list of the currently active semantic wikis.

3 Semantic Recommender Systems

In this section we briefly introduce how semantics can be taken advantage of in the context of recommender systems, and how it improves the results as they take into account the truly underlying reasons that determine the users satisfaction or dissatisfaction about the items.

Traditional Collaborative Filtering algorithms proceed by calculating similarities between users or between items [6]. These similarities are based on the ratings given to the items by the users. In the first case (user based), a user will receive recommendations made up of the items that similar users liked best. In the second case (item based), the recommended items will be those that are similar to the ones that the user loved in the past. This latter approach is known to be more efficient, since the similarities can be calculated off line [7].

If semantic features are taken into account, then the similarities could be computed according to them. This is what we call *Semantic Filtering Recommendation*. Indeed, the semantic features are the underlying reasons owing to which the items are similar or not. As we will see in section 4.5, our item-based approach for developing for a semantic recommender systems is based on domain ontologies containing the semantic attributes for the items. We use OWL ontologies and a reasoner able to classify the described resources.

4 TasTicWiki

TasTicWiki is a wiki engine that supports the creation and management of semantic wikis with recommendation services. This engine is born from the mixture between the ideas of semantic wikis and semantic recommender systems. Both of them are utterly better off with the addition of semantic annotations, and we have developed an architecture where this extra information can be used in an homogeneous way, both for the semantic wiki and the recommendation system sake. In fact, we can see the recommendation services as an enhancement of wiki search, which is purely one of the leading leitmotivs of adding semantics to wikis.

4.1 TasTicWiki architecture

In figure 1 we illustrate the architecture of TasTicWiki.

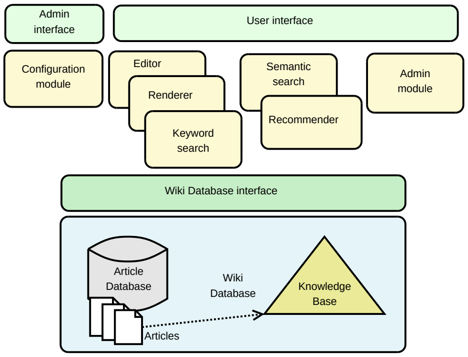


Fig. 1. TasTicWiki architecture

Every article in TasTicWiki is stored inside the database and it also corresponds to an instance inside the knowledge base, i.e., the ontology. The semantic metadata is thus stored separately from the page content, but we have set up a cache inside the database that will serve basic semantic information at the time of rendering and making certain type of queries, for the sake of reducing time processing. We use the knowledge base only when an article is firstly classified and when the users request queries involving complex axioms. The modules over the database interface are the ones that implement the functionality of the wiki. The *admin module* is devoted to administrative tasks such login, logout, registration, management of user profiles, etcetera.

Knowledge Base. TasTicWiki relies on a background ontology preloaded in the knowledge base. This domain ontology depends on the specific topic of the wiki. For example, we have developed a domain ontology in the field of tourism, since we have implemented a wiki[1] for a tourist information system. This background ontology has to fulfill some conditions in order to be used as a logic model in our knowledge base. It needs two main classes or concepts: one for storing the articles and another one for the different features the articles may have. We need

[1] http://khaos.uma.es/wikitrip

at least one role connecting the former with the latter, i.e., a *hasFeature* role -but nothing prevents the existence or others roles.

As an example, we briefly explain the skeleton of the tourism ontology we have developed. It has a *Tourist Service* class devoted to store the instances of the regular articles inside the wiki. These instances are related to the instances inside the class (or subclasses of) *Tourist Service Feature*, via some roles including *hasFeature* -we have three more roles as sub roles of the last one: *hasTradeActivity*, *hasSportActivity* and *hasSpecialty.* In addition, we count with some data roles establishing properties like the price, opening and closing times, etcetera.

In order to make the ontology expressive enough, we have defined some sub classes of the article class (i.e., the Tourist Service one). They are in much cases defined with complex axioms, e.g., there is a class called *Department Stores* which is defined as a service with at least two different trade activities. Another example can be *Inexpensive Accommodation*, defined as every *Accommodation Service* whose price is lower than thirty euros. The idea behind these definitions is that the Knowledge Base will perform some reasoning over the annotations the users include inside the text of the articles.

4.2 Semantic Annotation

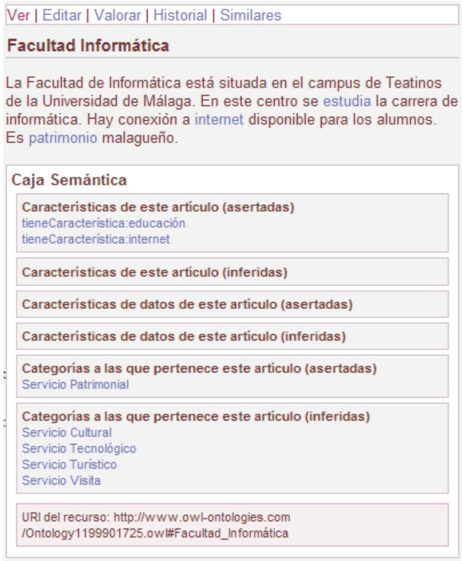
When creating or editing an article, users in TasTicWiki may include two kinds of semantic annotations. This is done by special wikitext commands, and they consist in: a) annotations about features and b) annotations about categories the article belongs to. In a), the system needs the user to specify both some role and some feature value (or equivalently, some data role and some data value). In b), only the name of the category is needed. In 4.6 we will show an example of the wiki text used to add this semantic annotations.

4.3 Enhanced navigation and presentation

When rendering an article, TasTicWiki provides a Semantic Box, which summarizes all the available semantic metadata. The kind of information present in the Semantic Box depends on the type of article that is being rendered. Indeed, inside TasTicWiki exists a clean classification of articles depending on their concrete roles, as we explain in the next section. In figure 2 we can see an article with its Semantic Box.

Types of articles. In this section we describe each type of article in TasTicWiki and some details about them.

- **Regular articles**: they are the standard articles of the wiki, i.e., those whose purpose is just spreading knowledge about some particular topic. In our example, they would be the articles standing for Tourist Services. Users are allowed to insert semantic annotations in their wikitext. The information contained in their Semantic Boxes are: asserted features and categories



Facultad Informática

La Facultad de Informática está situada en el campus de Teatinos de la Universidad de Málaga. En este centro se estudia la carrera de informática. Hay conexión a internet disponible para los alumnos. Es patrimonio malagueño.

Caja Semántica

Características de este artículo (asertadas)
tieneCaracterística:educación
tieneCaracterística:internet

Características de este artículo (inferidas)

Características de datos de este artículo (asertadas)

Características de datos de este artículo (inferidas)

Categorías a las que pertenece este artículo (asertadas)
Servicio Patrimonial

Categorías a las que pertenece este artículo (inferidas)
Servicio Cultural
Servicio Tecnológico
Servicio Turístico
Servicio Visita

URI del recurso: http://www.owl-ontologies.com
/Ontology1199901725.owl#Facultad_Informática

Fig. 2. An article inside the wiki, with its Semantic Box

(i.e., those explicitly specified by the users with semantic annotations) and inferred features and categories (those inferred by the reasoner).

- **Special articles**: they represent ontology entities like categories, feature concepts (in which we can found lists with feature values), feature values themselves, roles and data roles. Users are not allowed to add semantic annotations on them, but they can edit the wikitext in a pure textual way. Their semantic boxes show structural information like sub and super classes, domains and ranges, etcetera.

Users may create regular articles and feature value ones, but not the articles corresponding to concepts or roles. In other words, they are not allowed by the moment to edit the architecture of the background ontology (only their instances). This is considered as future work on the TasTicWiki system.

4.4 Enhanced search

Traditional wikis normally offer some kind of keyword, textual search. This sometimes is not powerful enough to retrieve the articles we need, as keywords do not really grasp the semantics underneath. In TasTicWiki we have developed a semantic search module, in which users, through a friendly, graphical interface, will be able to build and share complex queries based on complex ontological axioms.

It is not only about typical database search like *tell me all the services with a price lower than thirty.* It goes beyond, as complex axioms aim to recover articles following not only the explicitly provided information, but implicit knowledge as well. As we are working with OWL ontologies, these axioms are the ones who exist in OWL DL: cardinality restrictions, universal and existential quantifiers, value axioms, negation axioms and membership axioms, with logical connectives as glue. In figure 3 we can see the interface for building complex queries.

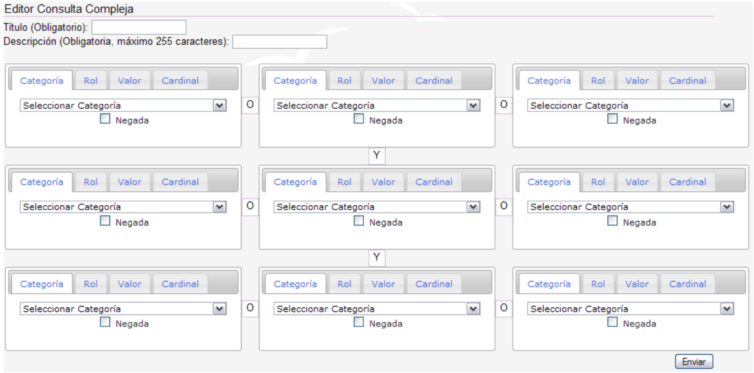


Fig. 3. Interface for complex queries. It is a matrix of atoms in which the user can specify some logic axioms that the obtained articles have to fulfill.

4.5 Recommendation

In Semantic Filtering Recommendation [10], we compute similarities between articles depending on the available semantic metadata. Then, given a set of well rated articles in the past, we compute the final recommendations. In next sections we go through the details of this process.

Analyzing users interactions. Inside a wiki we have several sources of information that can be taken into account at the time of computing the satisfaction of users. The direct one is collecting explicit ratings about the articles, asking for a numeric evaluation. But we also can take advantage of the previous behavior of the user inside the system: searchings, readings and editions. These last source is somewhat wiki-specific and, though by the moment is only used as a numeric coefficient (i.e., we only focus on the *quantity* of editions), an immediate future work way is taking into account the *quality* of the editions, mostly the semantic ones. This source of information could be used not only for the recommendations sake (e.g., we could infer semantic categories or features of which the user is a connoisseur), but also for supporting the edition tasks, offering suggestions of possible annotations that would go well with the current wiki text.

The *configuration module* allows the administrator of the system to decide a weighting coefficient of all these factors in order to compute the satisfaction degree that an article has for an user (e.g., we could consider that explicit ratings are more important than the rest of factors). We need this degrees in order to build the input for the recommendation algorithm described in the next section.

Recommendation process. Providing we have a set of articles that satisfied a given user to some extent, computed from the study of the past interactions that the user has performed inside the wiki as we explained in the previous section, we are now able to compute the final recommendations. Given a well-rated article, its neighborhood is the set of the n most similar articles in the system. The similarity between two articles is calculated as follows:

$$sim_{i,j} = \frac{|SIP(i) \bigcap SIP(j)|}{max(|SIP(i)|, |SIP(j)|)}$$

Where *SIP*(i) is the *Semantic Item Profile* of the item (article) i, calculated by means of the Article Ontology -i.e, it is the set of semantic categories the item i belongs to. Note that similarities range from 0 to 1.

Once we have computed all the neighborhoods of the well-rated articles, we recommend those items in the union of all the neighborhoods that fulfill the next two conditions: the article has not been read by the selected user and the *Recommendation Factor*, which is a measure of how good the recommendation will be for an user, is bigger than a certain number, called *Recommendation Threshold*[2]. The Recommendation Factor is calculated as follows:

$$RF(i) = r(father) * sim_{i,father}$$

Where *father* is the article from which the neighborhood was calculated. If an article belongs to more than one neighborhood, then we take into account the biggest factor of all the possible recommendations. The *Recommendation Threshold* that we use to filter the items depends on the ratings domain and could be parametrized, as well as the size of the neighborhoods -in terms of percentage of the total number of articles in the system.

4.6 An example

Let us imagine an user who is going to use the wiki for a while. We will see through a simple, brief example how this experience will be like. In `http://khaos.uma.es/wikitrip` we can find the concrete wiki used for this example, called Wikitrip, developed in the topic of tourism services inside Malaga, Spain.

[2] This threshold can go from 0 to the upper limit of the ratings, e.g., from 0 to 5

Editing an article. The user wants to create an article about a hotel where he stayed during his last holidays. It was a three-star hotel with lift, private bathroom and a price of forty euros. Moreover, he consider that its category is medium. Among other textual information, the user wants to specify this four semantic annotations, task which will be performed by special wikitext commands:

> ...Astoria Hotel has [[feat:hasFeature:with lift/lift]], [[feat:hasFeature:private bathroom]] and a price per night of [[dfeat:hasPrice:30]] euros. Is is a [[cat:Medium Category]] service and...

As we can see, special, different commands are used depending on whether we are specifying features (*feat*), categories (*cat*) or data features (*dfeat*).

Navigation and presentation. Once the user has saved this article, it will be presented with links in the places where the semantic annotations were inserted.

- For features (i.e., with *lift* and *private bathroom*) a link to the corresponding *feature value article* will be rendered.
- For categories (i.e., *Medium Category*) a link to the *category article* will be rendered.
- In the case of data features, it makes nonsense to render a link to the value *30 euros*. Instead, an special type of link is generated: a query of all the articles inside the wiki which have a price of 30 euros.

Inside the Semantic Box of the article the user will find the most interesting pieces of information. Here, the system shows the implicit information extracted from the semantic annotations the user has inserted. Specifically, we will find that the article belongs to four categories: one explicitly inserted by the user (Medium Category) and three inferred by the reasoner, this is: *Tourist Service*, *Accommodation Service* and *Inexpensive Service*.

The information about the features will not be rendered in the semantic box as links to articles. Instead, these links lead to special queries which retrieve all the articles in the system related to the same value through the same role. In the case that implicit feature relations are inferred, they will be shown inside the Semantic Box as well.

Searching. The user can do some searching inside the wiki. It could be in a pure textual way, as in many traditional wikis, but also in a semantic way. Thanks to the underlying reasoner and a proper interface, the user will be able to make queries like: All the Catering Services with either a price lower than thirty or with at least three different specialties. Once the result list is computed, the user can read, edit and rate the given articles.

Recommendations. Once the user has read, searched, edited or rated some articles inside the system, the recommendation module will be able to compute a list of recommended items as we explained in section 4.5. If the user does not have any experience inside the system, then this list will be made up of the most popular articles (measured by explicit ratings).

5 Related work

Fred Durao and Peter Dolog have proposed a tag-based recommendation [11] as an extension for KiWi [5], with three slightly distinct approaches which offer different levels of performance and quality. In the more complex approach, they compute similarities between articles according to the tags the users have used to annotate them. Basically, this system differs from ours in the sense that we use reasoning in order to compute the *tags* -categories, indeed- and they only rely on the users criteria. Nevertheless, they plan to develop some reasoning to infer semantic similarities between tags, but even in that case, our approach turns in another flavor, since we extract the tags or categories from the Knowledge Base. For example, if we have an article *a* talking about a catering service with a price of ten euros, and an article *b* about an accommodation service with a price of forty euros, our system will tag both of them with the concept *Inexpensive Service*, and we will use that information in order to compute the final recommendation.

6 Work status and Future work

TasTicWiki is currently in its beta version, providing the services we pointed out in previous sections. Some future work is actually needed: an internationalization module, some improvements in the edition interface -as well as taking advantage of semantics in the edition tasks-, OWL/RDF export, and of course, the possibility of editing the underlying ontology in a collaborative way.

Other issues like performance are also to be studied, since we are using rich, expressive ontologies that do not go well with complexity. Complex queries are hard to solve, and we need scalable reasoning able to respond within a tolerable time. DBOWL [12] is a persistent and scalable reasoner which stores the underline ontology using a relational database which could be integrated with our current repository, allowing the composition of complex queries with the right level of abstraction thanks to a special, ad-hoc query language.

Moreover, when complex queries are requested, we need the knowledge base to be prepared and adapted to every previous change in the annotations of articles. This means, in reasoning terms, that the underlying ontology needs to be classified regularly in order to show complete results, so more solutions in this field are to be investigated. Furthermore, an evaluation of the recommender system inside the wiki needs to be done.

7 Conclusions

TasTicWiki is a wiki engine that allows the creation and management of semantic wikis with recommendation services. Semantic metadata improves presentation and navigation inside the wiki. TasTicWiki relies on background, rich ontologies that make possible advanced reasoning tasks, improved searching and some sophisticated functionalities as content recommendation.

Acknowledgments. This work was supported by the ICARIA Project Grant, TIN2008-04844 (Spanish Ministry of Education and Science), the pilot project *Formación y desarrollo de tecnología aplicada a la biología de sistemas*, P07-TIC-02978 (Innovation, Science and Enterprise Ministry of the regional government of the *Junta de Andalucía*) and the project *Creación de un soporte tecnológico para un sistema de información turística*, FFI 055/2008 (Tourism, Sport and Commerce Ministry of the regional government of the *Junta de Andalucía*).

References

1. Wikipedia: Wiki. `http://en.wikipedia.org/wiki/Wiki#Characteristics` (2009)
2. Vlkel, M., Krtzsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic Wikipedia. Proceedings of the 15th international conference on World Wide Web. Pages:585 - 594 (2006)
3. Schaffert, S.: IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises. Pages 388 - 396 (2006)
4. Kuhn, T. AceWiki: AceWiki: A Natural and Expressive Semantic Wiki. Semantic Web User Interaction at CHI. (2008)
5. Schaffert, S., Eder, J., Grnwald, S., Kurz, T., Radulescu, M: KiWi - A Platform for Semantic Social Software (Demonstration). ESWC 2009: 888-892 (2009)
6. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. Knowl. Data Eng. 17(6): 734-749 (2005)
7. Mobasher, B., Jin, X., Zhou, Y.: Semantically Enhanced Collaborative Filtering on the Web. EWMF 2003: 57-76 (2003)
8. Rui-Qin Wang, Fan-Sheng Kong: Semantic-Enhanced Personalized Recommender System. International Conference on Machine Learning and Cybernetics. Volume: 7, On page(s): 4069-4074 (2007)
9. Linden, G., Smith, B., York, J.: Industry Report: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Distributed Systems Online 4(1): (2003)
10. Ruiz-Montiel, M., Aldana-Montes, J.F.: Semantically Enhanced Recommender Systems, On the Move to Meaningful Internet Systems: OTM 2009 Workshops, volume 5872/2009, pages 604 - 609 (2009)
11. Durão, F., Dolog, P.: Analysis of Tag-Based Recommendation Performance for a Semantic Wiki. 4th Semantic Wiki Workshop at the 6th European Semantic Web Conference, volume 464 (2009)
12. Roldán, M., Aldana-Montes, J. F. : Complete OWL-DL Reasoning Using Relational Databases. Database and Expert Systems Applications, volume 5690, pages 435 - 442 (2009)