Journal of Web Engineering, Vol. 13, No.1&2 (2014) 160-180

ACOTA: A MULTILINGUAL AND SEMI-AUTOMATIC COLLABORATIVE TAGGING WEB-BASED APPROACH

CÉSAR LUIS ALVARGONZÁLEZ

WESO Research Group, University of Oviedo, Oviedo, Spain
cesar.luis@weso.es

JOSE MARÍA ÁLVAREZ-RODRIGEZ

Knowledge Reuse Group, Carlos III University of Madrid, Leganés, Spain
josemaria.alvarez@uc3m.es

JOSE EMILIO LABRA GAYO PATRICIA ORDOÑEZ DE PABLOS

WESO Research Group, University of Oviedo, Oviedo, Spain
labra@uniovi.es patriop@uniovi.es

Received July 31, 2013
Revised December 20, 2013

This paper introduces a multilingual hybrid methodology to automatically deploy and combine collaborative tagging techniques based on user-behavior and recommendation algorithms. A reference web architecture called ACOTA (Automatic Collaborative Tagging) is also described in order to show the recommendation capabilities of this approach with the aim to assist users when multilingual resource tagging is required. Finally a quantitative research in the context of corporate knowledge management is also presented to evaluate and assess the goodness and accuracy of the methodology to minimize the effort of multilingual document categorization.

Key words: Collaborative Tagging; Automatic Tagging; Multilingual data; Document Retrieval; Knowledge Management
Communicated by: B. White & L. Carr

1 Introduction

The sheer mass of data already available in the Internet and the increasing use of mobile devices such as smart-phones, tablets or e-books readers are generating a new and dynamic data/information domain in which new requirements are emerging [1]. Some time ago information resources were printed on paper and kept in cabinets but nowadays this new vast amount of data is commonly stored in digital formats. To achieve this, different organizations and access techniques are used with the objective of facilitating processes such as information and document retrieval or search and report, to name a few. Although they try to take advantage of indexing and categorizing algorithms, on-line information resources usually contain a lot of heterogeneities [2] that cannot be easily addressed. More specifically, the multilingual encoding of information is one of the main drawbacks to provide a common access to the information [3]. In this sense, Internet contents are currently available in a huge variety of languages. Overall, English is the most common language [4], used in 54.9% of websites, and most of the tools and techniques to perform some kind of exploitation over this information are customized to work on it. This implies that the rest of information (45.1%) faces a lack of tools to properly exploit this information. If we consider that there are around 14.24 billion indexed web sites [5], 6.44 billion of web sites are expected to be in other languages rather than in English. As a consequence, new tools are required to bridge the gap between the exploitation capabilities of English and non-English information resources.

Phase	Year	Value
Size of the WWW (Number Websites)	14.24 billion	2013
Size of the WWW	2.5 exabytes	2013
Websites in English	2013	54.9%
Websites in Other Languages	2013	45.1%

Table 1 Statistical Data about the size of the WWW

Organizing this vast amount of multilingual data can be tedious, but in some enterprise and academics fields, it can become the cornerstone to success. E-learning systems, B2B environments, Corporate Knowledge Management or Extraction and Information Retrieval are some domains in which data filtering and management processes are becoming crucial. This is to obtain more accurate and timely information with the aim of enabling new business opportunities. More specifically knowledge organizations [6] and workers have gained their momentum in the new information society. Activities such as efficient document description, indexing or classification of information resources are now major challenges due to the aforementioned dynamic data domain. In this sense Hjørland [7] establishes different approaches, such as the use of controlled domain-based vocabularies and information retrieval techniques, user-oriented cognitive views or bibliometric approaches among others. This is carried out in to ease the tasks of knowledge workers. As a consequence, organizations with knowledge management activities can take advantage of discovering new business opportunities or workers capabilities due to the analysis of their daily activities. In this sense Technology Watching [8] is one of the key activities in some of these companies. For example, Treelogic S.L.[a] is a SME (Small and medium enterprise) technology company in Spain that has devoted a percentage of its working time to this task where workers add everyday news, blog posts, research articles, funding opportunities, amongst others. These elements help the company to be aware of new trends. In this environment the proper classification and exploitation of information resources leads to a knowledge organization that can exploit this information and, as a consequence, its know-how for improving their own performance.

[a] http://www.treelogic.com

In order to address these new requirements in knowledge organizations there are several well-known and traditional techniques to model, structure, organize and exploit data and information such as conceptual maps, taxonomies or controlled vocabularies. Nevertheless recently there has been a growing use of ontologies and folksonomies as a method to efficiently manage big amount of information resources. An ontology is usually defined as a specification of a shared conceptualization [9], it is a formal description of concepts and their relationships involved in knowledge domain that assist to organize and build knowledge-based system. On the other hand, a folksonomy [10] is the result of free categorization, without a defined structure or formality. Usually it is created by collective intelligence; a group of users with different levels of knowledge interested in some domain collect and classify information resources. These two emerging approaches have been successfully applied to specific domains such as e-Health, e-Government or e-Procurement (ontologies [11],[12],[13]) and Web 2.0 sites (folksonomies [14][15]) such as Delicious[b], Flickr[c] or YouTube[d].

In general, ontologies are more adequate to classify domain-specific information and data in a restricted context in which experts in a field have reached a common and shared understanding. In this context, data, information and knowledge is commonly concrete, static and stable. Obviously this approach presents some drawbacks to handle data, information and knowledge management in a more general and dynamic environment. Some structural upper ontologies such as DOLCE[e], SUMO[f] or PROTON[g] have been delivered in order to formalize high-level entities and relationships. On the other hand, as Clay Shirky pointed in [16], a folksonomy works better with a large and dynamic corpus, unstable entities and participants of all levels of knowledge. These characteristics seem to fit better in the web trying to exploit information to provide business intelligent services such as marketing suggestions or knowledge discovery.

In this paper, we present a hybrid methodology employing automatic tagging techniques and user-behavior recommendation algorithms to take advantage of folksonomies as a previous step to consolidate knowledge in an ontology. This methodology is built on top of several techniques such as natural language processing, querying to both folksonomies and ontologies, collaborative or machine learning techniques, delivering a multilingual methodology which does not suffer from cold start [17]. We have developed a software library called Automatic COllaborative Tagging (hereafter ACOTA) as a reference implementation that provides an automatic tagging engine with collaborative and user-behavior recommendation capabilities. Upon this library, we have also developed a web architecture. A quantitative evaluation taking into account existing information retrieval measures such as precision and recall is also performed over a real dataset (480 tagged information resources). This dataset was

[b] https://delicious.com

[c] http://www.flickr.com

[d] http://www.youtube.com

[e] http://www.loa.istc.cnr.it/DOLCE.html

[f] http://www.ontologyportal.org

[g] http://proton.semanticweb.org

created by the employees of the Treelogic S.L. a company with the objective of assessing the tagging capabilities of the ACOTA library.

The rest of the paper is structured as follows. Section 2 describes related work. Section 3 presents the proposed methodology. An overview of the architecture to support a methodology for multilingual and domain-less collaborative tagging is described in section 4. Section 5 presents the web architecture while section 6 presents a knowledge-management case study. Section 7 described the experiment methodology, evaluation and discussion and finally section 8 conclusions.

2 Related Work

Taking into account the use of blogs as information sources there are a number of works [18][19][20] that generate tags from these sources. Although knowledge information goes beyond blog-post tagging, it has a certain similarity, since a blog-post can be seen as traditional document, that covers many and specific different topics, rather than web sites. In this sense, Brooks et al. [18] built a system that was based on the use of the top three term frequency–inverse document frequency (TFIDF) score tags of a blog post. This approach groups documents into clusters with the hypothesis that a cluster of documents that share a tag should be more similar than a randomly constructed set of documents. The proposed methodology uses a similar approach but with a different purpose. Suggesting common tags that are within the same cluster for a given tag. AutoTag developed by Mishne et al. [19] is a tag suggestion blog-post engine which employs collaborative filtering methods. This tool offers suggestions for tags based on tags assigned to similar posts, leaving the editor the decision of choosing the proper set of tags. Our methodology also transfers the final decision to the user, and employs similar suggestion techniques, but it does not rely only on collaborative methods, adding a pre-automatic tag generation step, based on the structure of the document. Finally, TagAssist is another automatic tag suggestion engine developed by Sood et al. [20], it evolves AutoTag design, adding support for tag compression. This approach provides a good support for English posts that serves as inspiration for the proposed multilingual methodology but it is exclusive for this language, in contrast with our solution. Furthermore and due to the growing use of social networks, some works have emerged to detect experts [21], trends and filter information [22] among others. Our proposed approach takes advantage of these works to adapt broad techniques in a narrower context such as knowledge management in a company.

In the case of Automatic Keywords Extraction [23][24], Song et al. [25] proposed a clustering and classification based tag recommendation. This study uses a Poisson mixture model for document classification in addition to a novel node ranking method. Sun et al. [26] evolved it with a language model for tag recommendation (LMTR) approach, this technique compares how similar documents are to documents based on the shared words. Our research employs a slightly different approach than these works, it also compares how similar are documents but instead, taking into account the shared tags (previously selected by the users) from the documents. Dostal et al. [27] developed an automatic keyphrase extraction based on Natural Language Processing (NLP) and statistical methods. In this paper we employ more customized and domain-based NLP techniques.

Most of the existing works are focused on English tag recommendation. In the case of TagAssist [25] and Brooks [23] they drop out any non-English words; on the other hand AutoTag [24] takes a less aggressive solution, handling non-English words, but due to the application of an English stemmer for any word, regardless of the language, it ends up giving low scores to non-English posts. In the case of ACOTA, as a multilingual automatic and collaborative tagging engine, it is focused on providing support for non-English languages. Our motivating scenario was Spanish but one of our initial design requirements was not to restrict the system to only one language and to facilitate the configuration and the extension to other languages.

Another issue in recommendation systems is the cold start [17]. It appears when a recommendation system is relatively new or unused, so there is not enough data to properly perform the recommendation algorithms. This issue is partially addressed by our system and it is able to suggest comprehensive tags even when there is no data stored on the system given that part of the suggestion takes as input public existing folksonomies. In this sense, a similar approach has been followed in previous works [28][29][30][31] that have been made use of queries to folksonomies such as Google Complete API to enrich tag suggestion. Nevertheless, in the context of the present paper, results of empirical experiments show a considerable amount of noise that decreases the accuracy of the suggestions. For instance the delegation of suggestions to this service without any pre/post processing implies the generation of a vector of suggestions with *n* tags, where some tags are representative according to the initial query but they are not usually representative enough for a domain-based recommendation. In order to partially reuse this service and avoid non-representative tags our system pre-filter the queries and filters the external suggestions according to the existing domain.

Taking into account the main features and highlights of the aforementioned works, we have extracted the top features, which a domain less automatic collaborative tagging technique should have, see Table 2.

Feature
Multilingual
Collaborative
Domainless (No Specific Domain)
Not Suffering from Cold Start (No Training Required)
NLP Techniques
Querying to Ontologies
Querying to Folksonomies

Table 2 Recommender Criteria

3 A Multilingual - Domainless Automatic Collaborative Tagging Methodology

In this section, a detailed description of the proposed methodology is provided. The methodology is divided in two main stages: 1) the automatic tagging engine that is in charge of text normalization, keyword extraction, and tag enrichment and 2) the recommendation tagging engine, which uses the outcomes of the first stage to finally generate tag suggestions in a certain context.

3.1. Automatic Tagging Engine

The automatic tagging engine is also comprised of two phases. Firstly, the extraction phase which consists of tag extraction based on the document structure applied to an existing dataset. Afterwards, the previous results are enhanced in the enrichment stage, querying to both folksonomies and ontologies. As a result, a set of descriptive tags are generated by the automatic tagging engine. Before this process and as usual in any technique dealing with natural language, an optional but relevant step must be executed. This should be considered in order to clean the raw text, removing special characters, stop-words and so forth. The aim of this cleaning is to decrease the potential noise and to avoid the spreading of these non-useful lemmas to further stages.

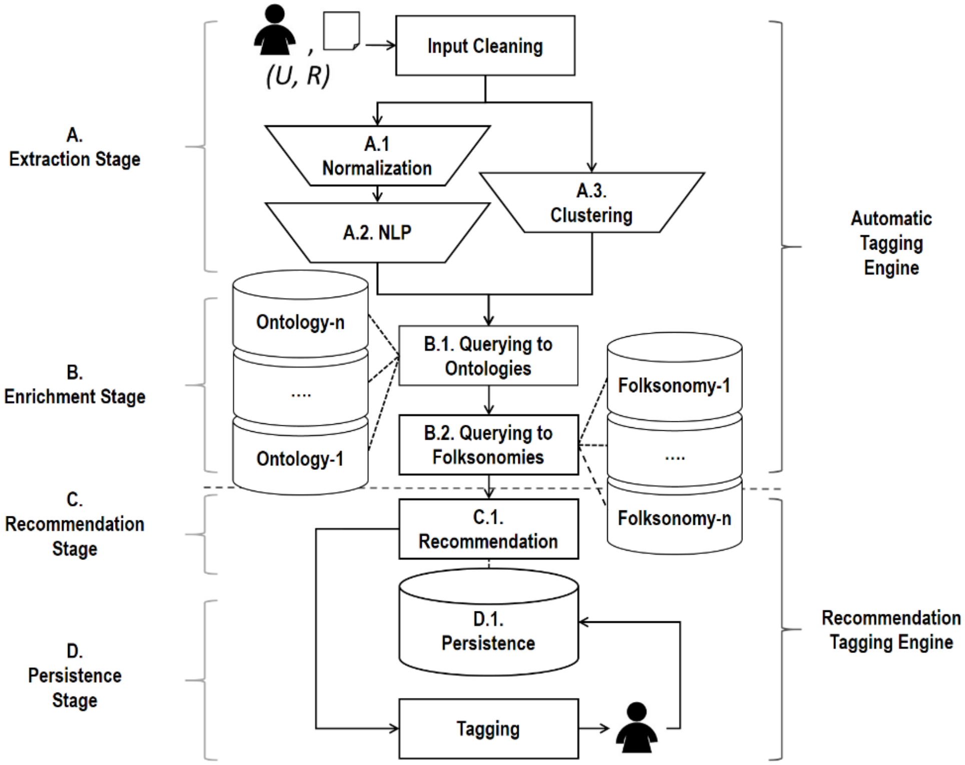


Figure 1 Workflow of a Multilingual - Domainless Automatic Collaborative Tagging Methodology

A. Extraction Phase. This phase extracts tags from a document. It is composed of two different extracting techniques: 1) unigrams generation, it is performed by normalization and NLP, and 2) *n*-grams generation, a refined combination of the aforementioned techniques.

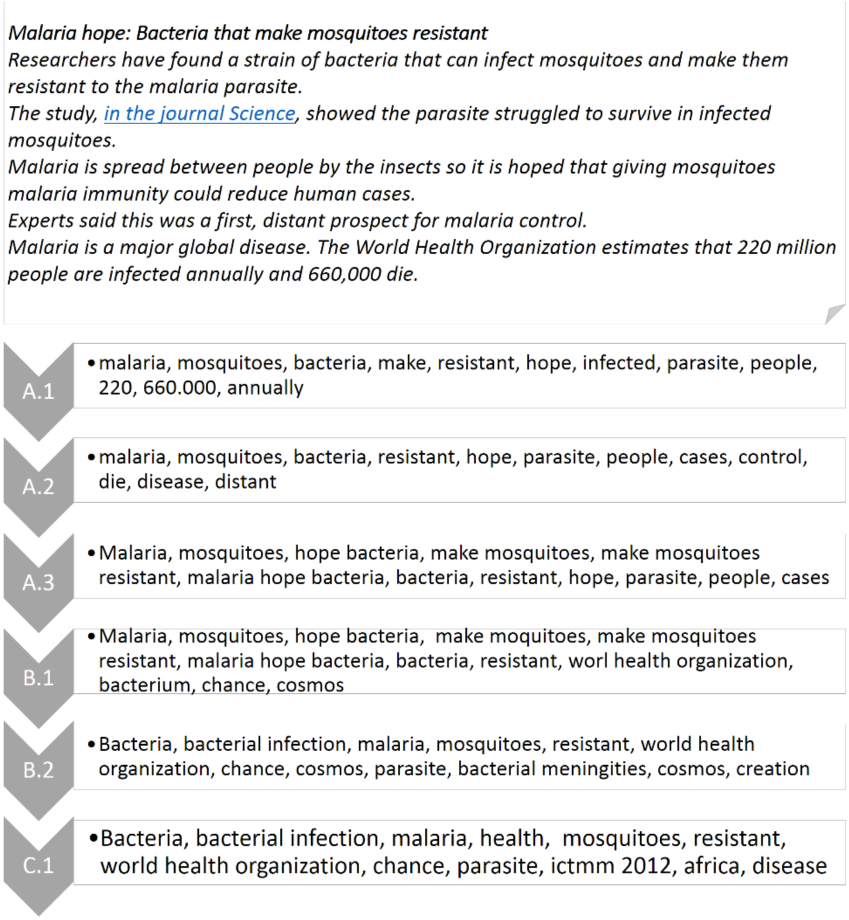
A.1. Normalization. This phase takes as input an information resource and produces a vector of unigrams. The first step is to split words at punctuation characters, removing punctuation marks and other delimiters. Then, each token is turned to a lower case. Tokens with less than three characters are removed as they are usually irrelevant for the results [31][32]. Finally, tokens are filtered according to a stop-word set (depending on the language common stop-words sets are available and can be configured), ending up with a clean set of unigram tags.

For each tag the number of occurrences is counted, multiplying this number by a weight. According to the word position or appearance, the weight is calculated. Thus words appearing in the title or specific fields [33] will have more weight than the ones presented in the body or raw contents.

A.2. Natural Language Processing. As Bohemia et al. [27] pointed out, keywords are usually formed by nouns, adjectives or present & past participles. This filtering technique consists on modifying the weight of a tag based on its morphosyntactic type.

Firstly, for each keyword, its Part Of Speech (POS) tag is calculated. Based on it, adjectives, nouns, and participles are stored according to their grammatical category. A small set of blacklisted POS tags are then removed while the rest continue to the next step. Finally the value of each tag is increased based on the vector in which they are stored.

A.3. Clustering. The combination of *Normalization* and *NLP* techniques provides a fast and simple method to suggest single-word tags. In some cases, unigrams cannot provide enough semantics, due to the simplicity of the meaning which can be supplied by an isolated word. As a consequence the use of *n*-grams, more specifically bigrams, or even short sentences are techniques that can be applied to improve the potential final set of suggestions.



Malaria hope: Bacteria that make mosquitoes resistant
Researchers have found a strain of bacteria that can infect mosquitoes and make them resistant to the malaria parasite.
The study, in the journal Science, showed the parasite struggled to survive in infected mosquitoes.
Malaria is spread between people by the insects so it is hoped that giving mosquitoes malaria immunity could reduce human cases.
Experts said this was a first, distant prospect for malaria control.
Malaria is a major global disease. The World Health Organization estimates that 220 million people are infected annually and 660,000 die.

A.1
- malaria, mosquitoes, bacteria, make, resistant, hope, infected, parasite, people, 220, 660.000, annually

A.2
- malaria, mosquitoes, bacteria, resistant, hope, parasite, people, cases, control, die, disease, distant

A.3
- Malaria, mosquitoes, hope bacteria, make mosquitoes, make mosquitoes resistant, malaria hope bacteria, bacteria, resistant, hope, parasite, people, cases

B.1
- Malaria, mosquitoes, hope bacteria, make moquitoes, make mosquitoes resistant, malaria hope bacteria, bacteria, resistant, worl health organization, bacterium, chance, cosmos

B.2
- Bacteria, bacterial infection, malaria, mosquitoes, resistant, world health organization, chance, cosmos, parasite, bacterial meningities, cosmos, creation

C.1
- Bacteria, bacterial infection, malaria, health, mosquitoes, resistant, world health organization, chance, parasite, ictmm 2012, africa, disease

Figure 2 Ongoing Example (Top-12 Tags)

The first step to generate *n*-grams consists in splitting the text into clusters from 1 to κ tokens. For each cluster, POS tags are generated. Next, they are analyzed from the edges, removing non-suitable tokens, until a valid one is found. This process is performed once for each edge, removing the whole cluster if all of the tokens are no-suitable. Finally, as in the normalization for each cluster the number of occurrences is counted, multiplying this number for a weight.

B. Enrichment Stage. The enrichment stage receives a vector of tags as payback from the extraction stage. In this stage, tags are enriched by making queries to both ontologies and folksonomies.

B.1. Querying to Ontologies. This technique consists on enriching tags obtained by suggesting synonyms querying to the WordNet [34] dictionary, version 3.0.

Each tag is looked up in WordNet, if the tag is in the dictionary, its synonyms are retrieved, if not, the stage is finished. Each synonym is checked if it is within the vector of tags, if this happens, the tag's weight is increased, otherwise the synonym is added to the vector of tags with a default value.

B.2. Query to Folksonomies. As previous sections have presented, querying folksonomies can be a double-edged sword, on the one hand this technique increases the amount of representative words; on the other hand it can be unsafe because it can easily increase the amount of noise in the system. Taking into account this critical point, the proposed methodology uses a chain of filters to enrich only those words over a given percentile (this parameter has as default value of 50 in order to skip the 50% less representative tags but it can be customized by the user in a latter feedback stage). Therefore, only representative tags are enriched, skipping those that do not provide an extra meaning. This filtering process, indirectly, helps to reduce the number of queries to external services, usually REST calls, reducing bandwidth consumption and execution time.

Improving performance: Caching REST Calls. An underlying problem in the on-going approach lies in the time consumption when external services are requested to enrich or extend some of the words. Due to the fact that a single tag enrichment requires, at least, one external REST call, when the number of tags is big enough, the execution time can be overkill, even with the filters employed by our approach.

In order to prevent this situation, a configurable cache has been added to anticipate and save previous results of REST requests, thus the system is able to avoid a big number of requests, reducing a lot of bandwidth consumption and execution time.

In the particular case of Intranets or enterprise environments, where the amount of processed documents is considerable, the use of a cache speeds up dramatically the process, given that a huge amount of words will be stored within the corporate cache, reducing querying time.

3.2. Recommendation Tagging Engine

The main aim of the *Recommendation Tagging Engine*, is to suggest tags based on the previous behavior of users.

C. Recommendation Stage. An item-based recommender algorithm has been selected for making suggestions. This algorithm recommends tags based on how similar items are to items [35], employing as similarity measure the Tanimoto Coefficient [36][37]. Furthermore the methodology has been designed to easily support the addition of new algorithms being technologically independent of the recommendation technique. The main idea is: if a set of tags, for instance "*wikipedia*" and "*encyclopaedia*" are tagged together on several documents, the recommendation engine would suggest "*encyclopaedia*" if "*wikipedia*" is presented and vice versa.

The main drawback of this approach is that when the system is "*cold*" [17], the recommendation engine cannot work properly. In a pure recommendation engine, this can become a serious problem due to the sole reliance on these techniques. That is why a "*hybrid*" approach combining the automatic tagging engine with the recommendation tagging engine has been designed. Therefore, when the suggestion engine is cold, the automatic tagging engine still works underneath, providing results.

Example. The word "*canary*" has at least three different meanings, a domestic and colorful bird, the Spanish Archipelago in the northwest coast of Africa and a development version of the popular web browser, chrome[h]. In different scenarios, the recommendation engine should behave suitably to them, recommending tags based on their meaning according to the system's context. For instance in an e-learning system used for biology lessons, it would recommend tags related to that field, such as "*bird*", "*animal*" or "*pet*". In a travel agency, it would recommend tags like "*islands*", "*tourist*" or "*Spain*" and finally in a technology blog it would suggest tech tags such as "*Chrome*", "*web browser*" or "*Google*".

D. Persistence Stage. Once a set of tags is presented to the final user, she is able to decide whether to pick some of the provided tags or to add new ones that are stored as feedback for future recommendations.

4 Description of the ACOTA System

We developed a library called ACOTA (Automatic Collaborative Tagging). It consists of two main components, the Core Component which provides the Automatic Tagging Engine, and the Feedback Component, which provides the "collaborative" and "recommendation" capabilities. Both, Core and Feedback components can be used as standalone projects or can be combined together to take advantage of the features provided by each module.

The two components are comprised by a set of enhancers, each enhancer relates to a specific methodology's step, as it is shown in Table 3. An enhancer consist in a set of custom features built upon several tools and/or libraries.

Phase	Stage	ID	ACOTA Component
Automatic Tagging engine	Extraction	A.1	Core
Automatic Tagging engine	Extraction	A.2	Core
Automatic Tagging engine	Extraction	A.3	Core
Automatic Tagging engine	Enriching	B.1	Core
Automatic Tagging engine	Enriching	B.2	Core
Recommendation Tagging Engine	Recommendation	C.1	Feedback
Recommendation Tagging Engine	Persistence	D.1	Feedback

Table 3 Mapping between the proposed methodology and ACOTA

[h] https://www.google.com/intl/en-419/chrome/browser/canary.html

4.1. Core Component

This component enables and supports Automatic Tagging Engine, it includes a set of Enhancers which provides features as term extraction, natural language processing and querying to both ontologies and folksonomies.

A. Extraction Stage. The extraction stage is performed by the combination of *LuceneEnhancer*, *OpenNLPEnhancer* and *TokenizerEnhancer* implementing *Normalization*, *NLP* and *Clustering* phases, respectively.

A.1. Normalization. LuceneEnhancer implements the normalization stage and it extracts unigrams from an information resource, normalizing the result. This *Enhancer* is implemented using Apache Lucene[i] in order to process the input text, removing stop-words and punctuation, among others.

A.2. Natural Language Processing. The *Natural Language Processing* phase is implemented by *OpenNLPEnhancer*. It uses the Apache OpenNLP[j] library to translate words into POS tags. This POS tags are used in order to apply filtering techniques which modifies the tag's weight, based on the morphosyntactic type of the tag.

A.3. Clustering. TokenizerEnhancer implements the *Clustering* phase, it employs more advanced natural language processing and extraction techniques in order to extract n-grams. As POS tag translator, this enhancer has also been implemented using the Apache OpenNLP library.

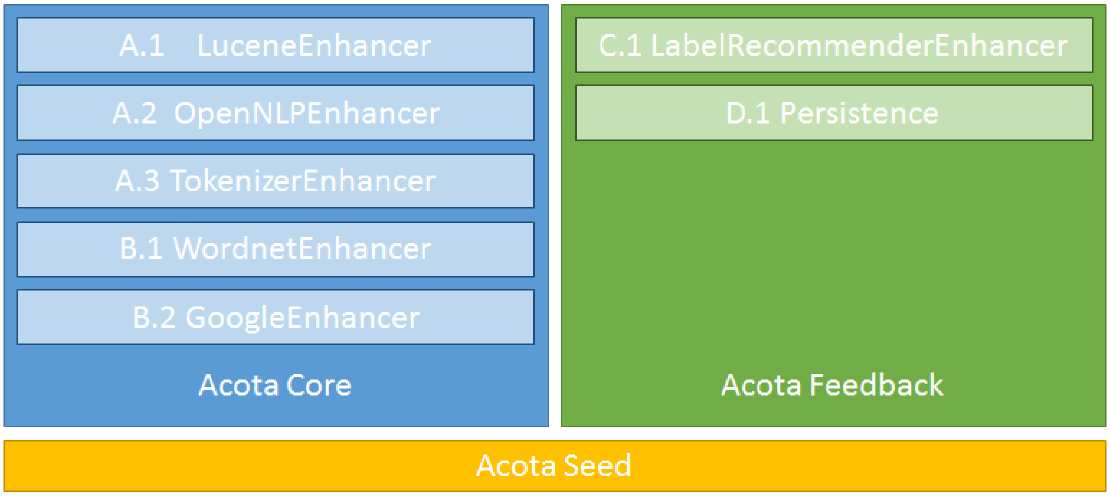


Figure 3 ACOTA's Architecture

B. Enrichment Stage. This stage consist in making queries to ontologies and folksonomies, in order to enrich the set of tags provided by the extraction stage. This stage uses two enhancers: Wordnet (*WordnetEnhancer)* and the Google Complete API (*GoogleEnhancer).*

[i] http://lucene.apache.org

[j] http://opennlp.apache.org

B.1. Querying to Ontologies. It requires the WordNet dictionary to provide synonyms for English tags. It employs the Java Wordnet Interface[k] (JWI) capabilities to mount dictionary files directly on-disk, thus English tags are processed by these Enhancer in an isolated mode. Due to the flexibility of this implementation the support for new languages can be easily added configuring the Wordnet dictionary.

B.2. Querying to Folksonomies. It uses Google Complete[l] service as a folksonomy, enriching the results with new tags. This folksonomy, suggest the top ten most used queries on Google, by a given set of words, in our case the tag to enrich is used as the query.

As it has been previously explained, the prototype employs an optional caching system, in order to reduce bandwidth consumption and execution time. As caching system we have employed Memcached[m], a high-performance and distributed memory object caching system.

4.2. Feedback Component

This component corresponds with the Recommendation Tagging Engine and it is divided in two sub-components, a recommendation engine which allows the system to recommend tags based on previous users behavior within the system, and a persistence system, which stores the users' feedback, enabling improvements in the accuracy of the recommendation engine.

C.1. Recommendation Engine. The *LabelRecommenderEnhancer* implements the Recommendation Stage, it suggest tags based on previous users behavior. This Enhancer is implemented using Apache Mahoutn, a machine learning library which includes a predefined set of algorithms. The current prototype uses a customized version of the Mahout's item-based recommenders, with the aim of taking advantage of a well-known technique, focusing on recommendations and employing documents as users and tags as items.

D.1. Persistence. Existing machine learning libraries require a persistence system based on different formats. In the case of Mahout, previous data is stored in a relational database, more specifically in a table containing the tuple (document, item, weight). The system has been designed to avoid database-lock in and other vendors or storage systems can be easily plugged to ACOTA such as MySQL[o], MariaDB[p] and PostgreSQL[q] and also NOSQL systems like MongoDB[r].

4.3 Extending ACOTA

[k] http://projects.csail.mit.edu/jwi

[l] https://www.google.es/

[m] http://memcached.org

[n] http://mahout.apache.org

[o] http://www.mysql.com/

[p] http://mariadb.org/

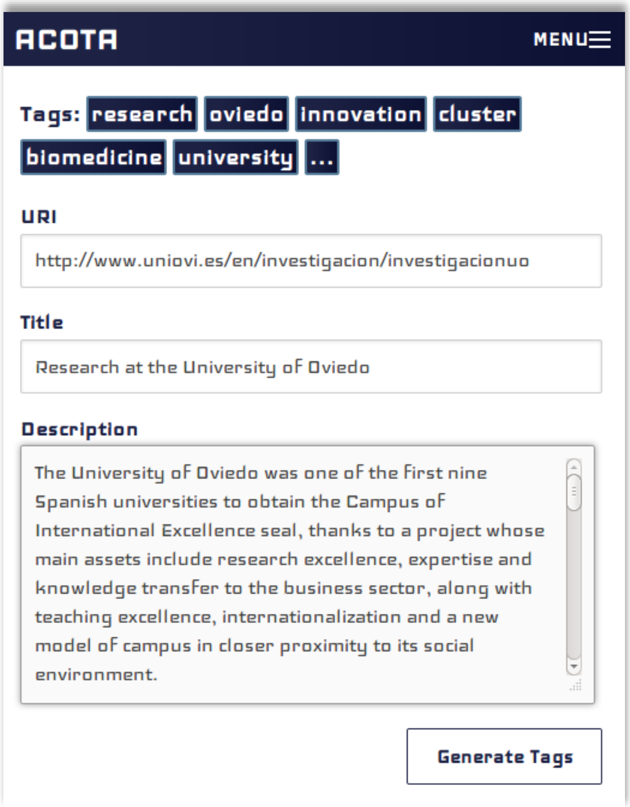
[q] http://www.postgresql.org/

[r] http://www.mongodb.org/

ACOTA was built taking extensibility into account; therefore it can be easily extended in two different ways. On the one hand it can be extended by adding support to other languages. This can be performed configuring the proper OpenNLP and/or WordNet files for the desired languages. On the other hand ACOTA can be extended creating new enhancers that supports novel functionalities by extending ACOTA-seed.

4.4. ACOTA in action

There is an available demo of ACOTA running at Heroku[s], see Figure 4, in the following URL http://acota.herokuapp.com . This demo generates a set of tags from a body and a title inserted by the user. It also enables the modification of parameters used by default. In addition to this, the web site includes information and tutorials about how to use ACOTA as a library or as a WebSocket Service. It is also important to emphasize that it is an open source project under the Apache 2.0 license and can be located in: https://github.com/weso/acota-{component} (where component is: *core*, *seed*, *feedback*, *utils and web*).



Tags: research oviedo innovation cluster biomedicine university ...

URI

http://www.uniovi.es/en/investigacion/investigacionuo

Title

Research at the University of Oviedo

Description

The University of Oviedo was one of the first nine Spanish universities to obtain the Campus of International Excellence seal, thanks to a project whose main assets include research excellence, expertise and knowledge transfer to the business sector, along with teaching excellence, internationalization and a new model of campus in closer proximity to its social environment.

Figure 4 Screen capture of ACOTA demo.

[s] https://www.heroku.com/

5 Turning ACOTA into a Web Architecture

In this section we present the conversion performed to turn ACOTA, a non-web library, into a web application.

ACOTA is comprised by six different steps, each single step receives as entry the output of the previous one. The amount of time required by each step is quite short, but B.2. Querying to Folksonomies *and* C.1. *Recommendation Engine* may take an extra time, depending on the number of cached tags and/or the number of records within the database. Therefore, the implementation of ACOTA must take these points into account.

5.1. REST API vs. WEBSOCKET API

The first question we had to face was how users are supposed to consume the ACOTA web API. Users will send a triple (URI, title and content) and the API answers a set of tags comprised by a triple (tag, weight, language). We considered two possibilities for the API implementation: a REST based API and a WebSocket API.

Representational State Transfer is a pattern of resource operations that has emerged as a de facto standard for service design in Web 2.0 applications [38]. The first implementation (R.1) used a single REST call to process all the steps, but as the processing of all the steps may take some time, the user would have to wait until the tags were sent back. From a user interface (UX) point of view, making the user waiting is not practical, since it breaks the "ensuring visibility of system status" heuristic [39]. Therefore we designed a second solution (R.2), which separates each step process in an independent rest call and solves the waiting drawback. However, since the amount of data shared in each petition can be large (every single response can be comprised by several thousand of tags), it implies a bandwidth and time penalty. A third approach (R.3) would be to store in the server-side all the temporal results, but it breaks the stateless constraint of REST architectures [40].

Approach	Technology	UX Constraint	REST Constraint	Bandwidth Consumption	Elapsed Time
R.1	REST	-	+	+	-
R.2	REST	+	+	-	-
R.3	REST	+	-	+	+
W.4	WebSocket	+	N/A	+	+

Table 4 Studied Approaches to turn ACOTA into a Web Application

WebSocket [41] is a technology which allows the system to keep an open connection between a client and a server, so both actors can send and receive asynchronous messages. This feature is really useful for real time applications.

The final approach (W.4) takes advantage of WebSockets, so the server sends the results back to the client as they are computed. This approach keeps the "ensuring visibility of system status" heuristic, since the client receives information in real-time, updating the results as they are received. The use of WebSockets also reduces the size of the shared data, since the whole bunch of tags are kept in the server, and it just sends the top-12 tags of each step back to the client. Therefore the last approach suits our needs as can be seen in Table 4.

5.2. Actor Model & ASK Pattern

Since ACOTA-WEB employs WebSockets, and there is a need to handle asynchronous messaging, we employed an actor model architecture [42] as the one used in the Erlang[t] programming language or the akka[u] framework. The present web-architecture is fairly simple: 1) Once the client sends a message with the triple (URI, title and content), the server side instantiates an actor which process every step sequentially. 2) Firstly the message-timestamp is stored within an actor which works as a shared memory object. 3) The timestamp is spread from one step to another with the aim of checking if the current actor is the last one instantiated (If it is not, it means that the client has sent a new triple, so the current actor is dismissed). 4) Once a step is computed and it is checked that there are no new messages, the results are sent back to the user. This checks are crucial, since it works as a lock, checking that the messages sent back to the client are the ones provided by the most recent triple.

6 A Corporate Knowledge Management case of study

Treelogic S.L. is a Spanish company which provides customers with information and communication technology-based solutions. Treelogic S.L. has a solution called *Imaginn Watching*, a research and development (R&D) technological watching tool. The aim of this solution is to be up to date with the latest trends in technology and business opportunities within their sector.

Technological watch consists [8] in “watching” regularly similar areas such as legal, social, technological or environmental in order to have the company’s internal information up to date and ready to be consulted by the company’s decision-makers.

Feature	ACOTA	Brooks	AutoTag	TagAssist	Dostal
Multilingual	+	-	-, partial	-	-
Collaborative	+	+	+	+	+
Domainless (No Specific Domain)	+	-, Requires Training	-, Requires Training	-, Requires Training	+
Not Suffers from Cold Start (No Training Required)	+	-	-	-	-
NLP Techniques	+	-	-	+	+
Querying to Ontologies	+	-	-	-	-
Querying to Folksonomies	+	-	-	-	-

Table 5. Collaborative Tagging Technique for assisting corporate knowledge management criteria

This kind of tool requires handling and organizing vast amounts of data that usually comes from the Internet sources in a variety of languages. The domain of the data is weak and is constantly changing as new terms, technologies and business opportunities are coined regularly [43]. All this

[t] http://www.erlang.org

[u] http://akka.io

information is created and consumed by workers from the different areas in the company in a collaborative way.

These technological-watch requirements fit with the multilingual-domain less automatic collaborative tagging features described in Section 2. Multilingual support, collaborative capabilities, the possibility to enrich the data with external sources of information (such as folksonomies, ontologies and so forth) and the lack of domain specific, cold start and the necessity of training the tagging engine, are the requirements that Treelogic S.L. have looked to fulfill with ACOTA. Table 5 shows how ACOTA and the previous studies satisfy the aforementioned features.

7 Research Evaluation

7.1. Design of the Experiment

In order to evaluate the results we use two different evaluation criteria, precision and recall [44]. Precision (1) is the fraction of retrieved tags that are relevant, in this case, the fraction of proposed tags which matches with the tags tagged by the real users.

$$Precision = \frac{Match}{|\{Recommended\ Tags\}|} = \frac{|\{Dataset\ Tags\} \cap \{Recommended\ Tags\}|}{|\{Recommended\ Tags\}|} \quad (1)$$

Recall (2) is the fraction of relevant tags that are successfully retrieved.

$$Recall = \frac{Match}{Count} = \frac{|\{Dataset\ Tags\} \cap \{Recommended\ Tags\}|}{|\{Relevant\ Tags\}|} \quad (2)$$

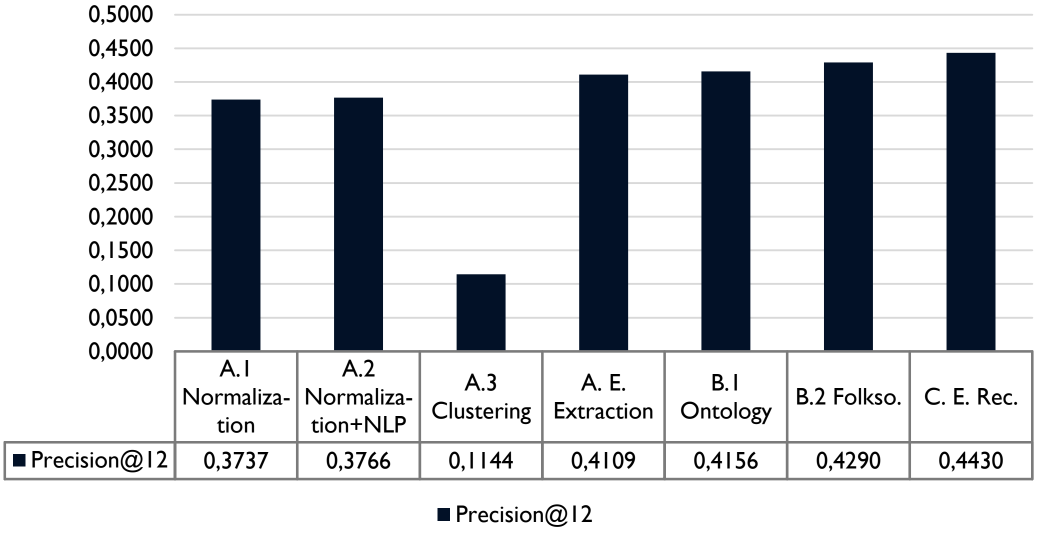
In both evaluation criteria we have employed the top-12 tags from the suggestions vector. Measuring precision at 12 retrieved results is referred as precision@12 (read ‘precision at 12’) and measuring recall at 12 retrieved results is referred as recall@12 (read ‘recall at 12’). Following a similar Mishne et al. [19] approach, it was assumed that if users do not find an accurate tag within the top-12 tags, they would skip the search and probably they would probably end up self-adding a new one.

7.2. Sample

We have employed a private dataset comprised by a set of 483 tagged documents and 1049 tags. This dataset has been provided in the context of a research project by the company Treelogic S.L. It is a slice of its research & development technological watching system and it has been generated by its workers in daily activities. These documents are both in English and Spanish, however the number of Spanish documents is bigger than in English. These information resources are in a variety of lengths and formats (e.g. plain text, HTML & XML) including typical data heterogeneities (e.g. special characters). The data was not pre-processed or cleaned trying to simulate the whole real scenario.

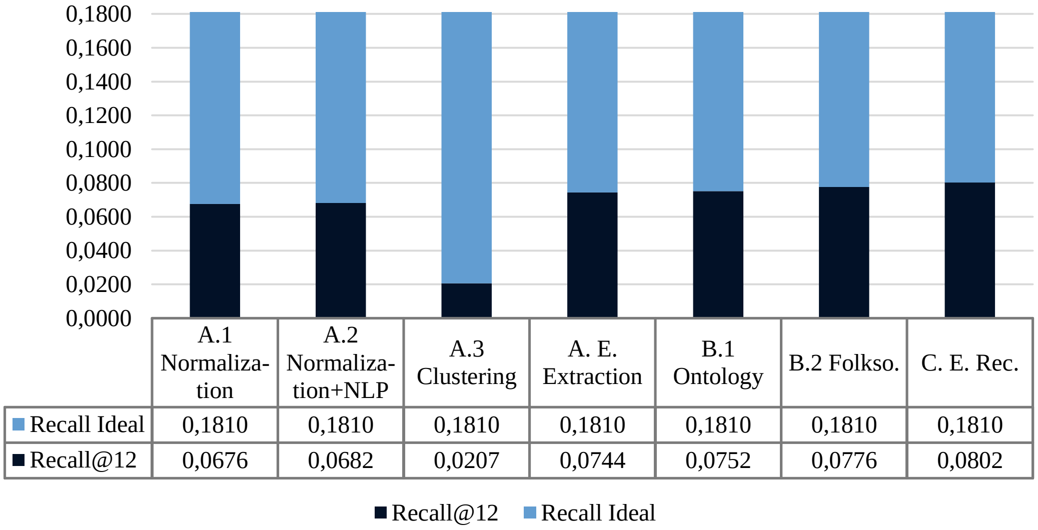
7.3. Results and Discussion

The precision@12 are presented in Figure 5 and the recall@12 statistics in Figure 6.



	A.1 Normalization	A.2 Normalization+NLP	A.3 Clustering	A. E. Extraction	B.1 Ontology	B.2 Folkso.	C. E. Rec.
■ Precision@12	0,3737	0,3766	0,1144	0,4109	0,4156	0,4290	0,4430

Figure 5 Precision@12 Values



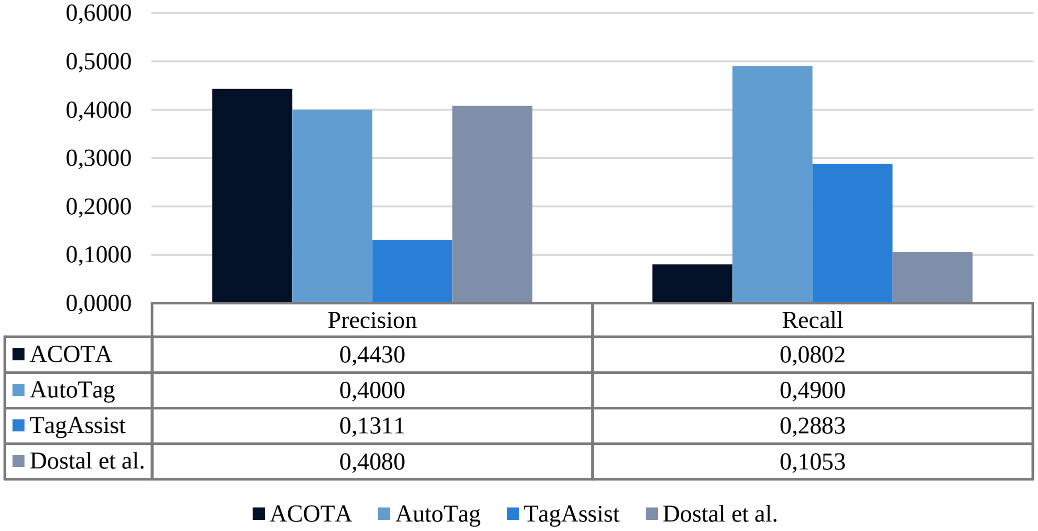
	A.1 Normalization	A.2 Normalization+NLP	A.3 Clustering	A. E. Extraction	B.1 Ontology	B.2 Folkso.	C. E. Rec.
■ Recall Ideal	0,1810	0,1810	0,1810	0,1810	0,1810	0,1810	0,1810
■ Recall@12	0,0676	0,0682	0,0207	0,0744	0,0752	0,0776	0,0802

Figure 6 Recall@12 Values

A. Extraction Stage As it was expected, the results show that the combination of Normalization + NLP and Clustering obtains the highest precision and recall values. Since the test data is comprised by unigrams and *n*-grams and these techniques are focused on recommending unigrams and *n*-grams respectively. The whole set of tags cannot be matched by each separated technique, due to the composition of the test data.

B. Enrichment Stage. The dataset contains a small percentage of English documents, as a consequence, querying to ontologies has almost no effect on the results. In contrast, querying to folksonomies increases dramatically both precision and recall.

C. Recommendation Stage. Starting from an empty database, in order to compute this value, as the satisfactory matches were found, they were used to feedback the system according to a chronological order with aim of simulating the real user's behavior. Thus the outcome of this stage serves to increase the number of positive matches and therefore the accuracy of future recommendations avoiding the cold-start.



	Precision	Recall
■ ACOTA	0,4430	0,0802
■ AutoTag	0,4000	0,4900
■ TagAssist	0,1311	0,2883
■ Dostal et al.	0,4080	0,1053

Figure 7 Relative precision and recall Values

It is worth mentioning that the test data contain a certain proportion of tags which make reference to temporal events, internal projects or even personal matters. Therefore, the number of measurable tags are slightly reduced, and even a very large folksonomy such as Google Complete API is not able to suggest these unmatched tags.

As it can be appreciated in both figures, recall@12 has low values. Due to the fact that the dataset-provided contains 1049 tags for 483 documents, the average of tags per documents is near 2.1718. Thus, the ideal recall@12 value would be 0.1810 (1049 tags tagged by experts per 5796 tags suggested by our system, 12 by each document). Therefore, the recall values are quite accurate, reaching values of 44.31%.

The precision and recall values obtained by ACOTA are quite moderate, so we decided to compare them with the results of the studied works (taking the best precision & recall values of their studies), see Figure 7. On the one hand the precision values are over the average of the studied works, on the other hand the recall values are below the average but even in this case, since the ideal recall value (0.1810) is too low, it is still also lower than the aforementioned average results.

8 Conclusions and Future Work

We have proposed a methodology for tagging multilingual documents in a semiautomatic way, employing extraction, enrichment and user-behavior recommendation techniques. Our methodology transfers the final decision of choosing the proper tag to the user, so if the recommended tags are non-suitable to him, he would add a new one giving feedback to the system. This feedback allows the recommendation engine to improve the results based on the previous behavior of the users.

We have also implemented a reference system called ACOTA. It was tested against production data from a research and development technology watching tool. Despite the considerable amount of internal and temporal tags, valid results have been obtained. This methodology and its implementation help knowledge workers to minimize the categorization-act effort providing a tool for better information resources classification within a knowledge organization.

As future work, we have considered increasing the amount of native languages, since English and Spanish are now supported; new languages can be added in order to internationalize the methodology and ACOTA as much as possible.

Another issue that we have taken into account is to improve the performance of the system. Although the system has a good performance with regular documents, it suffers when real-time (in terms of milliseconds) suggestions must be done processing large documents that contain several thousands of words.

Acknowledgements

ACOTA is a subproject of the ORIGIN project, co-financed by ERDF (European Regional Development Fund), with the aim of increasing the productivity of software development activities in global scenarios. This work has also been partially funded by Spanish project MICINN-12-TIN2011-27871, ROCAS (Reasoning on the Cloud by Applying Semantics).

References

1. Belkin, N., & Croft, W. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM, 29*(10), 1–10. Retrieved from http://dl.acm.org/citation.cfm?id=138861
2. Shklar, L., Sheth, A., Kashyap, V., & Shah, K. (1995). InfoHarness: Use of automatically generated metadata for search and retrieval of heterogeneous information. In *Advanced Information Systems Engineering* (pp. 217–230). Retrieved from http://link.springer.com/chapter/10.1007/3-540-59498-1_248
3. Large, A., & Moukdad, H. (2000). Multilingual access to web resources: an overview. *Program: electronic library and information systems, 34*(1), 43–58. doi:10.1108/EUM0000000006938
4. W3Techs. (n.d.). Usage of content languages for websites. Retrieved May 03, 2013, from http://w3techs.com/technologies/overview/content_language/all
5. Kunder, M. de. (n.d.). The size of the World Wide Web (The Internet). Retrieved May 03, 2013, from http://www.worldwidewebsize.com/
6. Hjørland, B. (2007). Semantics and knowledge organization. *Annual Review of Information Science and Technology*, *41*(1), 367–405. doi:10.1002/aris.2007.1440410115

7. Hjorland, B. (2012). Methods for evaluating information sources: An annotated catalogue. *Journal of Information Science*, *38*(3), 258–268. doi:10.1177/0165551512439178

8. Davidson, C. H. (2001). Technology watch in the construction sector: why and how? *Building Research & Information*, *29*(3), 233–241. doi:10.1080/09613210010027756

9. Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, *43*(5-6), 907–928. doi:10.1006/ijhc.1995.1081

10. Wal, T. Vander. (2007). Folksonomy Coinage and Definition. Retrieved from http://vanderwal.net/folksonomy.html

11. Casado-Lumbreras, C., Rodríguez-González, A., Álvarez-Rodríguez, J. M., & Colomo-Palacios, R. (2012). PsyDis: Towards a diagnosis support system for psychological disorders. *Expert Systems with Applications*, *39*(13), 11391–11403. doi:10.1016/j.eswa.2012.04.033

12. García-Crespo, Á., Rodríguez, A., Mencke, M., Gómez-Berbís, J. M., & Colomo-Palacios, R. (2010). ODDIN: Ontology-driven differential diagnosis based on logical inference and probabilistic refinements. *Expert Systems with Applications*, *37*(3), 2621–2628. doi:10.1016/j.eswa.2009.08.016

13. Villazón-Terrazas, B., Ramírez, J., Suárez-Figueroa, M. C., & Gómez-Pérez, A. (2011). A network of ontology networks for building e-employment advanced systems. *Expert Systems with Applications*, *38*, 13612–13624. doi:10.1016/j.eswa.2011.04.125

14. Hotho, A., Jäschke, R., Schmitz, C., & Stumme, G. (2006). Information retrieval in folksonomies: Search and ranking. *The Semantic Web: Research and Applications*, *4011*, 411–426. doi:10.1007/11762256_31

15. Yoo, D., Choi, K., Suh, Y., & Kim, G. (2013). Building and evaluating a collaboratively built structured folksonomy. *Journal of Information Science*. doi:10.1177/0165551513480309

16. Shirky, C. (2005). Ontology is Overrated: Categories, Links, and Tags. *Economics & Culture, Media & Community*. Retrieved from http://www.shirky.com/writings/ontology_overrated.html?goback=.gde_1838701_member_179729766

17. Park, S.-T., Pennock, D., Madani, O., Good, N., & DeCoste, D. (2006). Naïve filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’06* (pp. 699–705). New York, New York, USA: ACM Press. doi:10.1145/1150402.1150490

18. Brook, C. H., & Montanez, N. (2006). Improved annotation of the blogopshere via autotagging and hierarchical clustering. *Proceedings of the 15th World Wide Web Conference (WWW06)*. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Improved+Annotation+of+the+Blogopshere+via+Autotagging+and+Hierarchical+Clustering#0

19. Mishne, G. (2006). AutoTag. In *Proceedings of the 15th international conference on World Wide Web (WWW 06)* (p. 953). New York, New York, USA: ACM Press. doi:10.1145/1135777.1135961

20. Sood, S. C., Owsley, S. H., Hammond, K. J., & Birnbaum, L. (2007). TagAssist: Automatic Tag Suggestion for Blog Posts. In *ICWSM*. Boulder, Colorado, US. Retrieved from http://www.icwsm.org/papers/paper10.html

21. Noll, M. G., Au Yeung, C., Gibbins, N., Meinel, C., & Shadbolt, N. (2009). Telling experts from spammers. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09* (p. 612). New York, New York, USA: ACM Press. doi:10.1145/1571941.1572046

22. Hong, L., Ahmed, A., & Gurumurthy, S. (2012). Discovering geographical topics in the twitter stream. *Proceedings of the 21st international conference on World Wide Web*, 769–778. doi:10.1145/2187836.2187940

23. Gayo-Avello, D., Álvarez-Gutiérrez, D., & Gayo-Avello, J. (2004). Naïve Algorithms for Keyphrase Extraction and Text Summarization from a Single Document Inspired by the Protein Biosynthesis Process. *Biologically Inspired Approaches to Advanced Information Technology*, *LNCS 3141*, 440–455. doi:10.1007/978-3-540-27835-1_32

24. Mika, P., Ciaramita, M., Zaragoza, H., & Atserias, J. (2008). Learning to Tag and Tagging to Learn: A Case Study on Wikipedia. *IEEE Intelligent Systems*, *23*(5), 26–33. doi:10.1109/MIS.2008.85

25. Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.-C., & Giles, C. L. (2008). Real-time automatic tag recommendation. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08*, 515. doi:10.1145/1390334.1390423

26. Sun, K., Wang, X., Sun, C., & Lin, L. (2011). A language model approach for tag recommendation. *Expert Systems with Applications*, *38*(3), 1575–1582. doi:10.1016/j.eswa.2010.07.075

27. Dostal, M., & Ježek, K. (2011). Automatic keyphrase Extraction extraction based on NLP Automatic Keyphrase based on NLP and statistical methods and Statistical Methods. *Proceedings of the Dateso 2011: Annual International Workshop on DAtabases, TExts, Specifications and Object*, 140–145.

28. Labra Gayo, J. E., de Pablos, P. O., & Cueva Lovelle, J. M. (2010). WESONet: Applying semantic web technologies and collaborative tagging to multimedia web information systems. *Computers in Human Behavior*, *26*(2), 205–209. doi:10.1016/j.chb.2009.10.004

29. Jimenez-Nácero, W., Luis-Alvargonzález, C., Abella-Vallina, P., Alvarez-Rodríguez, J. M., Labra-Gayo, J. E., & Ordoñez de Pablos, P. (2012). Emergent Ontologies by collaborative tagging for Knowledge Management. In *Advancing Information Management through Semantic Web Concepts and Ontologies* (p. 16). IGI-Global.

30. Kern, R., Granitzer, M., & Pammer, V. (2008). Extending Folksonomies for Image Tagging. In *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services* (pp. 126–129). IEEE. doi:10.1109/WIAMIS.2008.43

31. Chen, P.-I., & Lin, S.-J. (2010). Automatic keyword prediction using Google similarity distance. *Expert Systems with Applications*, *37*(3), 1928–1938. doi:10.1016/j.eswa.2009.07.016

32. Sigurd, B., Eeg-Olofsson, M., & van Weijer, J. (2004). Word length, sentence length and frequency - Zipf revisited. *Studia Linguistica*, *58*(1), 37–52. doi:10.1111/j.0039-3193.2004.00109.x

33. Suchanek, F. M., Vojnovic, M., & Gunawardena, D. (2008). Social tags. In *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08* (p. 223). New York, New York, USA: ACM Press. doi:10.1145/1458082.1458114

34. Miller, G. a. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41. doi:10.1145/219717.219748

35. Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the tenth international conference on World Wide Web - WWW '01* (pp. 285–295). New York, New York, USA: ACM Press. doi:10.1145/371920.372071

36. Lipkus, A. (1999). A proof of the triangle inequality for the Tanimoto distance. *Journal of Mathematical Chemistry*, *26*, 263–265. Retrieved from http://link.springer.com/article/10.1023/A:1019154432472

37. Tanimoto, T. T. (1958). *An elementary mathematical theory of classification and prediction* (p. 10). International Business Machines Corporation (IBM), New York.

38. Battle, R., & Benson, E. (2008). Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST). *Web Semantics: Science, Services and Agents on the World Wide Web*, *6*(1), 61–69. doi:10.1016/j.websem.2007.11.002

39. Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Conference companion on Human factors in computing systems - CHI '94* (p. 210). New York, New York, USA: ACM Press. doi:10.1145/259963.260333

40. Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine. Retrieved from http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

41. I. Fette, Google, I., Melnikov, A., & Ltd., I. (2011). *RFC 6455 - The WebSocket Protocol* (p. 71). Retrieved from http://tools.ietf.org/html/rfc6455

42. Agha, G. (1985). *ACTORS:: a model of concurrent computation in distributed systems*. Retrieved from http://dspace.mit.edu/handle/1721.1/6952

43. Keats, J. (2010). *Virtual Words: Language on the Edge of Science and Technology*. Oxford, New York: Oxford University Press, Inc.

44. Cleverdon, C. W., Mills, J., & Keen, M. (1966). *Factors determining the performance of indexing systems* (Vol. I, p. 120). Cranfield. Retrieved from http://hdl.handle.net/1826/862