NASS: A Semantic Annotation Tool for Media[⋆]

Angel L. Garrido[1], Oscar Gómez[1], Sergio Ilarri[2] and Eduardo Mena[2]

[1] Grupo Heraldo - Grupo La Información. Zaragoza - Pamplona, Spain.
{algarrido, ogomez}@heraldo.es
[2] IIS Department, University of Zaragoza, Zaragoza, Spain.
{silarri, emena}@unizar.es

Abstract. Nowadays media companies have serious difficulties for managing large amounts of news from agencies and self-made articles. Journalists and documentalists must face categorization tasks every day. There is also an additional difficulty due to the usual large size of the list of words in a thesaurus, which is the typical tool used to tag news in the media.

In this paper, we present a new method to tackle the problem of information extraction over a set of texts where the annotation must be composed by thesaurus elements. The method consists of applying lemmatization, obtaining keywords, and finally using a combination of Support Vector Machines (SVM), ontologies and heuristics to deduce appropriate tags for the annotation. We carried out a detailed evaluation of our method with a real set of changing news and we compared out tagging with the annotation performed by a real documentation department, obtaining really promising results.

Keywords: Semantic tagging and classification; Information Extraction; NLP; SVM; Ontologies; Text classification; Media; News.

1 Introduction

In almost every company in the media industry, activities related to categorization can be found: news production systems must filter and sort the news at their entry points, documentalists must classify all the news, and even journalists themselves need to organize the vast amount of information they receive. With the appearance of the Internet, mechanisms for automatic news classification often become indispensable in order to enable their inclusion in web pages and their distribution to mobile devices like phones and tablets.

To do this job, medium and big media companies have documentation departments. They label the news they can, and the typical way to do that is by using thesauri. A thesaurus [1] is a set of items (words or phrases) used to classify things. These items may be interrelated, and it has usually the structure of a hierarchical list of unique terms.

[⋆] This research work has been supported by the CICYT project TIN2010-21387-C02-02.

We have worked with real data owned by publications of Spanish media companies associated to the Vocento[3] Media Group. These companies have a documentation department and they use *EMMA*[4] as their archive platform. News in EMMA are represented as records in a relational database. Every record includes the text of the article. These articles are tagged every day by the documentation department of these companies. The tagging consists of filling several fields in every record, and one of the most important is the thesaurus field. In that field, documentalists can write as many thesaurus terms as they want from EMMA's thesaurus hierarchical tree of terms. This is an assisted process, but anyway it takes much time because of the high number of news produced every day. It is a tedious work subject to a lot of human errors and it is very subjective, so it is very easy not to be rigorous. This is especially true when different people with different opinions are working together with a list of near one thousand thesaurus terms. Our goal is to help them do this daily and difficult job.

In this paper, we focus on the inner workings of the NASS system (*News Annotation Semantic System*), which provides a new method to obtain thesaurus tags using semantic tools and information extraction technologies. The seminal ideas of this project have been recently presented under a condensed format [2] and they have received a good acceptance. With this paper, we want to delve deeper into the operation of the system and to show the results in more detail.

In our system we first propose to obtain the main keywords from the article by using text mining techniques. At the same time, using Natural Language Processing (NLP) [3], the system retrieves other type of keywords called *named entities.* Second, NASS applies Support Vector Machines (SVM) [4] text classification in order to filter articles. Third, it uses the keywords and the named entities of the filtered texts to query an ontology about the topic these texts are talking about. Then, NASS uses the answers to those queries to increase the probability of obtaining correct thesaurus elements for each text, and it updates that matching score in a table. Finally, the system looks at this table and selects the terms with a score higher than a given threshold, and then it labels the text with the corresponding tags. We have tested this method over a set of thousands of real news published by a media company. As we had the chance to compare our results with the real tagging performed by the documentation departments, we have benefited from this real-world experience to evaluate our method [6].

This paper is structured as follows. Section 2 explains our method in detail. Section 3 discusses the results of our experiments with real data. Section 4 is a brief explanation of the state of art with SVM and Ontology-Based Information Retrieval. Finally, Section 5 provides our conclusions and some lines of future work.

[3] Vocento is a multimedia group in Spain, consisting of over 100 companies. For further information, please see *http://www.vocento.com/en.*

[4] EMMA is the Spanish abbreviation of *MultiMedia Environment Archiver*, which is a proprietary solution developed by Ibercentro Media.

2 NASS Methodology

In this section, we present our approach. First, we provide an overview of the architecture, and then we focus on the most interesting steps during the document annotation.

2.1 The System Architecture

Methods to obtain suitable tags for a text have been studied in the context of Information Extraction (IE). According to Russell and Norvig [7], IE means automatically retrieving certain type of information from natural language text. They say that IE is halfway between Information Retrieval (IR) systems and text understanding systems. However, according to a deep study of these issues recently performed by Wimalasuriya and Dou [5], works like ours could be classified in the field of Ontology-Based Information Extraction (OBIE), an emergent subfield of IE. Thus, it complies with the features identified in [5]:

- The system is going to process semi-structured natural language texts.
- The output is presented by using *ontologies*. An ontology is defined as a formal and explicit specification of a shared conceptualization [8], and consists of several components such as classes, data type properties, object properties, instances, property values of the instances, and constraints.
- The system uses an information extraction process guided by an ontology.

The main elements in an OBIE system are a preprocessor that works over the incoming text, an information extraction module (usually guided by a semantic lexicon like WordNet [9] and by a human-made ontology), and finally a Knowledge base used for storing the system's response. The architecture of our solution which is shown in Figure 1, fits in the general schema presented in [5] as seen later.

2.2 Implementation of the Proposal in a Real Context

In this section, we position our system in the production time line of a real media company. After finishing the everyday production, the news are obtained from the production system. Then, they are processed by EMMA and they are sent to a relational database. In that very moment the system has to produce tags, before documentalists access EMMA. There are some metadata already available, such as the date, section and author. Of course NASS also has the text and it is even able to identify different parts: the title, subtitle, introduction, signature, and other elements. However, there are several requirements to consider in the development of our system: the tags we obtain must belong to the documentation thesaurus, we have to use free software to develop our approach, and the tags should be obtained as fast as possible.

The outline of our method is as follows. First, NASS obtains from the text the names of the main characters, places and institutions, and then it deduces which

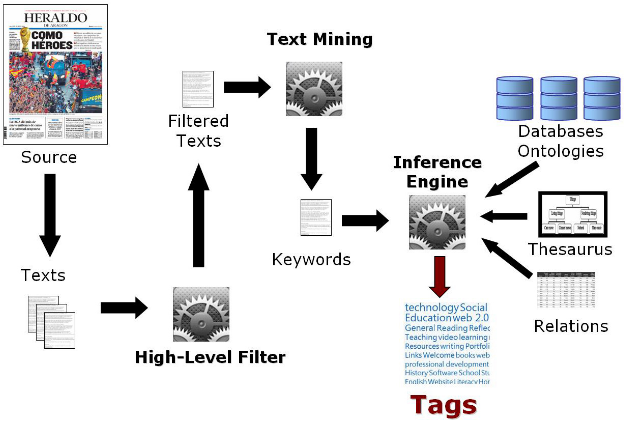


Fig. 1. General architecture of the NASS System.

the key ideas and major themes are. Second, the system uses this information in order to find related thesaurus terms. Finally, NASS assigns the thesaurus terms (with their ancestors) to the text. It looks easy, as basically it represents the way a person does this job. Unfortunately, making this task automatic is not that simple, as software does not understand text and lacks information about the context. So, we must look for some strategies that make it possible to obtain coherent terms from the thesaurus:

- The system has to obtain relevant keywords and named entities from the text, by using appropriate algorithms and a language analyzer.
- As soon as NASS obtains a list of keywords and named entities, it has to deduce their nature and find related thesaurus terms, by using SVM text categorization, a knowledge base, and a heuristic method.

Once the appropriate thesaurus terms have been identified, relating text with them and with their ancestors is trivial, as the system only has to traverse the thesaurus tree from the corresponding selected term upwards to the root. So, in the next subsections we will focus on the two first aforementioned tasks.

2.3 Lemmatization and Assignment of Keywords

Before obtaining keywords, NASS is going to lemmatize all the words in the text. So, lemmatization is the first step in the process, which means grouping together the inflected forms of a word. A lemma can be defined as the canonical form representing each word. This process simplifies the task of obtaining keywords and reduces the number of words the system has to consider later. It can also help to obviate *stop words*: prepositions, conjunctions, articles, numbers and other meaningless words. Moreover, it also provides us clues about the kinds of words appearing in the text: nouns, adjectives, verbs, and so on. For this purpose, we have used Freeling [10]. Freeling is an open source suite of language analyzers developed at TALP Research Center, at BarcelonaTech (Polytechnic University of Catalunya). After the lemmatization process, the system has a list of significant words, which is the input to the next process: obtaining keywords.

We propose merging two methods to obtain a set of significant keywords from a given text. The first method that NASS applies is a simple term frequency algorithm [11], but with some improvements. We call it *TF-WP* (*Term Frequency – Word Position*), which is obtained by multiplying the frequency of a term with a position score that decreases as the term appears for the first time towards the end of the document. This heuristic is very useful for long documents, as more informative terms tend to appear towards the beginning of the document. The TF-WP keyword extraction formula is as follows:

$$TF - WP = (\frac{1}{2} + \frac{1}{2} * \frac{nrWords - pos}{nrWords}) * TF$$

$$TF = \frac{nrRepetitions}{nrWords}$$

where $nrWords$ is the total number of terms in the document, *pos* is the position of the first appearance of the term, TF is the frequency of each term in the document, and *nrRepetitions* is the number of occurrences of that term.

The second method the system uses is the well-known *TF-IDF* (*Term Frequency – Inverse Document Frequency*) [12], based on the word frequency in the text but also taking into account the whole set of documents, not only the text considered. The TF-IDF keyword extraction formula is:

$$TF - IDF = TF * log_{10}(\frac{nrDocs}{D})$$

where TF is the frequency of each term in the text (as in TF-WP), *nrDocs* is the total number of documents, and D is the number of texts that contain that word. We have merged both methods by adding the two values TF-WP and TF-IDF after applying them a weight α and β, respectively. At the end of this task NASS obtains a list of keywords with their number of repetitions and weights.

2.4 Identification of Named Entities

In news, it is very common to find names of people, places or companies. These names are usually called *named entities* [13], which share a common feature: they start in uppercase. For example, if the text contains the words "Dalai Lama" both words could be considered as a single named entity to capture its actual meaning. This is a better option than considering the words "Dalai" and "Lama" independently. To do this task we have used Freeling too, which uses this identification method and provides a confidence threshold to decide whether to accept a named entity or not. In our system, this threshold is set at 75% in order to ensure a good result. NASS retrieves the more relevant named entities identified, replaces whitespaces by underscores (for example, "Dalai_Lama"), and finally adds them to the same collection of keywords obtained before.

2.5 Text Categorization

At this point, NASS has a list of the most important keywords and named entities in the input article. It could tag the text with this information, but we want to take a step forward by using the thesaurus for tagging. The question now is: how could NASS obtain thesaurus terms from a list of keywords? Performing a simple text comparison is not enough. For example, "SOUTH_AMERICA" is a thesaurus term used in documentation departments, but if an article is about *Argentina* it is unlikely that the words "South America" will appear. That article could speak about places or people in Argentina, and the keywords could be for instance *Argentina, Buenos_Aires, Cristina_Fernandez.* With that information, the system has to deduce that the thesaurus term desired is only "SOUTH_AMERICA". So, to be able to label that article with the correct thesaurus term, the system must know that Argentina is a country that belongs to South America or that Cristina Fernandez is the president of Argentina.

There are a lot of interesting ways to infer and deduce terms according to their meaning and their relationships with other terms. We have chosen SVM text categorization because it is a powerful and reliable tool for text categorization [4]. Regarding the type of SVM used, we have used a modified version of the Cornell SVM-Light implementation [14] with a Gaussian radial basis function kernel and the term frequency of the keywords as features [15].

Anyway, we have discovered some limitations as soon as we apply SVM over real news sets. SVM has a strong dependence on the data used for training. While it works very well with texts dealing with highly general topics, it is not the case when we need to classify texts on very specific topics not included in the training stage, or when the main keywords change in the text over time. For example, when we talk about sports, every year sportsmen may belong to different teams in the same competition. If we want to use SVM we will need to change the training set at least every year. This means that the documentation department should manually label a lot of articles (for training) before the system could tag new articles properly. Therefore, we use SVM to categorize texts within high-level topics, but we also change the strategy later to obtain more detailed tags.

We have improved the SVM results by using techniques from Ontological Engineering [16]. We advocate the use of knowledge management tools (ontologies, semantic data models, and inference engines) due to the benefits that they can provide in this context. The first step is to design an ontology that describes the items we want to tag, but this is not an easy task in media business. The reason is that media cover many different themes, and therefore it is a disproportionate task to try to develop an ontology about all the publishable topics. We think that a better approximation is to design an ontology for every interesting subject we want to tag, whenever SVM results are not able to get good results. In our experiments, we have selected the sports section, one of the most interesting for the audience, and at the same time one that has greater variability in terms of keywords from year to year. Inside this section, we have selected soccer articles as our target and we try to tag them automatically as best as the documentation department would do it. Therefore, our ontology includes teams, players, coaches, presidents, competitions, referees, and so on. Nevertheless, it is not only an ontology composed by named entities, as we have also introduced relationships and actions that join concepts providing semantic information, which is a substantial difference that brings advantages not contemplated by SVM. We have designed the ontology with the tool *Protégé* [18]. The ontology has been populated with current and rich data of the Spanish premier soccer league and stored in an OWL [19] file.

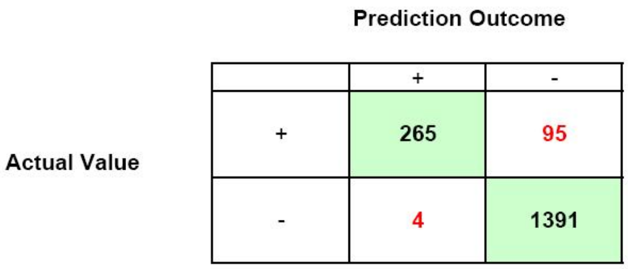
NASS tries to match each keyword with one of the words in the ontology. Before, we have prepared a table with the help of the documentation department and based on experimental and statistical analysis of the tags introduced manually. This table has two columns. In the first column we put ontology concepts. In the second column we put the probability of talking about a topic usually labeled with a term of the thesaurus when the system detects that concept in a text. Then, NASS submits SPARQL [20] queries against the ontology and it uses Jena [21] as framework and Pellet [22] as inference engine. As soon as it finds a keyword that matches a term of the ontology, it looks at its associated concept and then the system uses the previous table to retrieve the corresponding probability. At this point, it is important to mention that some keywords could be related with one or more thesaurus tags, and also a thesaurus tag could be related to one or more keywords. NASS increases the probability of tagging the article with a term each time it accesses a row of this table. A high number of accesses to the same thesaurus term guarantees that it can be used as a tag on the article. Through an extensive experimental evaluation we found useful to use 60% as a threshold to accept a term. Finally, NASS returns the thesaurus tags obtained by this method in order to label the text.

3 Experimental Evaluation

In this experimental evaluation, our goal is to evaluate the ability of the system to automatically annotate articles. All our experimentation has been performed with real sets of news belonging to a Spanish Media associated with the Vocento Group. We think the use of a standard corpus like the Reuters 25178, used in others approaches (for example, in [23] or [17]), is not so useful for us, as our objective is to solve a real problem with an own thesaurus and a particular way of labeling in a real documentation department. So, in our experiments we have used a corpus of 1755 articles, all of them tagged with thesaurus terms manually assigned by the documentation department (we will use them here in order to compare them with the automatic annotation performed by NASS). We consider that the input data set is a good representative of the types of articles managed.

We have applied lemmatization to the news corpus and then we have obtained keywords and named entities using the same value (0.5) for the parameters α and β (described in Section 2.3). Then, we have used SVM in order to find out the main topic of each article (for instance, to identify soccer articles). We trained NASS with thousands of sports articles from different years (already annotated). As one of the most important themes is the coverage of news related to soccer, which is the major sport in country, we refined the process applying NASS methods to decide which thesaurus terms were the best for labeling soccer articles. At this point, it is important to emphasize that the goal of our experimental evaluation was not only to detect which articles were talking about that topic, but also to properly label such articles by considering the suitable terms in the thesaurus.

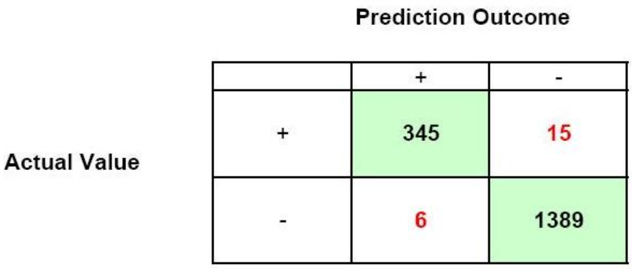
We compared the tags obtained by our system with those that were manually assigned by the people working in the documentation department. The results are shown in Figure 2 and Figure 3. Specifically, in Figure 2 we consider a poorly populated ontology and in Figure 3 a properly populated one (with an increase of 25% of the terms). The results are presented using two confusion matrices that represent the items that are correctly tagged on the existing 1755 texts using the same ontology but with a different number of elements. We will obtain from these matrices two interesting facts: precision and recall. Generally in pattern recognition and information retrieval, precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. In our case we will assimilate “retrieved” to ”properly tagged.”



		Prediction Outcome	
		+	-
Actual Value	+	265	95
	-	4	1391

Fig. 2. Experimental results after using NASS with a poorly populated ontology

From the previous figures, several conclusions can be drawn. First of all, we found that the highest number of well-labeled articles occurs when we suitably populate the ontology, and if NASS fails it is due to a lack of semantic information (for example, when the name of a coach or a team president has not been introduced in the ontology). When the ontology has fewer elements, then the recall drops significantly (73% vs 95%) but the precision is the same (98%); indeed, it even improves due to the smaller chance of error because NASS gen-



		Prediction Outcome	
		+	-
Actual Value	+	345	15
	-	6	1389

Fig. 3. Experimental results after using NASS with a properly populated ontology

erates a smaller number of labels. Anyway, the precision is good enough in both cases. Looking at the accuracy to put the tags, we could say that almost 99% of the labels generated by NASS were correct regardless of the number of elements that populate the ontology.

Summing up, the results obtained were really good. We obtained more than 95% of recall and precision. Furthermore, our system was able to detect additional labels that are relevant (even though they were not selected in the manual annotation) and avoid labels that were wrongly chosen by the documentation department.

4 Related Work

As commented along the paper, among other techniques, we have used a method commonly used for text categorization: the Support Vector Machine, or SVM [24]. This method has some appropriate features to face text classification problems, for example its capability to manage a huge number of attributes and its ability to discover which of them are important to predict the category of the text after a training stage. It is based on solid mathematical principles related to statistical learning theory and there are plenty of articles and books based on such mechanisms, not only for text classification but for any work related to cataloging all kinds of elements and entities. Our use of SVM is based on the references cited in Section 2.5.

Although OBIE is a relatively new field of study, most researchers believe that it could contribute very much to IE. Many researchers work with this kind of systems obtaining good results. So, in the last ten years, we can find an increasing number of articles facing IE problems by using OBIE systems. One of the earliest ones was *KIM* [25] (Knowledge Information Management). It was a generic solution and included an ontology, a knowledge base, a semantic annotation tool, and an indexing and retrieval server, as well as front-ends for interacting with the server. The developers of KIM used a corpus composed of newspaper articles and the system extracted information using linguistic rules and gazetteer lists. Another early work came from Maedche et al. [26], with an OBIE system including an IE system able to adapt itself according to the ontology used. It used partial parse trees as the information extraction method. Another approximation is Embley's work [27], where heuristics and parsers are used to improve the information extractor using car advertisements as corpus. Regarding our problem, all these papers have been useful to a greater or lesser extent, but we have missed a real tagging over the data to compare the results with the desires of the future users of the system. McDowell and Caffarella [28] annotated web pages with their tool *OntoSyphon*, but with an ontology-guided process instead of going over the set of documents. That is an interesting point of view but without application for us due to the way the data come in newspapers: our system is clearly a document-pivoted categorization, instead of a category-pivoted categorization.

As an example of project combining SVM and ontologies we can reference [29], a recent work where SVM is used to categorize economic articles using multi-label categorization. The big difference is that they use ontologies to create labels prior to the categorization process, and then use different types of SVM only in that process, which does not let them obtain neither a high degree of accuracy nor a high number of categories, whereas our proposal avoids these problems. We would also like to mention the work performed by Wu et al. [30], which faced this kind of problems using a quite interesting unsupervised method based on a Naive-Bayes classifier and natural language processing techniques. They obtained good results and it has the advantages of being an unassisted process, but they do not meet the minimum requirements of precision and accuracy that were needed for our project.

5 Conclusions and Further Work

In this paper, we have presented an information extraction system that helps documentation departments of media companies in their daily annotation job when they use a thesaurus as the annotation tool. To evaluate the accuracy of the system, we have performed experiments on real news previously tagged manually by the staff of the documentation department. This work has contributed to make a good adaptation of the general methods of SVM and OBIE systems over a real and changing set of news instead of the typical and less realistic standard news corpus. Our experimental results show that we are able to get a reasonable number of correct tags using IE methods like SVM, but the accuracy improves with the combined use of NLP and semantic tools when the training set of the SVM must be updated frequently (e.g., in the context of sports, where the specific data change every season). Instead of having to label news each year to create a reliable set for training, we propose that documentalists fill the instances of classes in a predefined ontology. Then, our system has enough information to label the news automatically by using semantic tools. We found this simpler and more intuitive for end users, and it helps to get better results.

Besides, the accuracy of the automatic assignment of tags with our system is very good, obtaining 99% of correct labels. In fact, the current version of NASS is already being successfully used in several companies. This shows the interest of our approach.

As an evaluation scenario, we have considered so far news related to sports. In the future, we will continue applying NASS to other news sections to verify its operation. We also plan to introduce new and more powerful methods to enrich the system providing it with greater speed, wider scope and better accuracy. Priority for us is also reducing the need for manual intervention during the process, an aspect in which we are already working. One of the most important challenges in this area is to develop a real, intelligent, useful and efficient semantic tool in the always-changing environment of the media, ready to help categorization tasks in archives and useful to distribute news in all the Internet formats: web, phones, tablets, and so.

References

1. A. Gilchrist. 2003. Thesauri, taxonomies and ontologies: an etymological note. Journal of Documentation vol. 59(1): pp. 7-18.
2. A. L. Garrido, O. Gomez, S. Ilarri and E. Mena. 2011. NASS: news annotation semantic system. Proceedings of ICTAI 11, International Conference on Tools with Artificial Intelligence: pp. 904-905. IEEE.
3. A. F. Smeaton. 1997. Using NLP or NLP resources for information retrieval tasks. Natural Language Information Retrieval. Kluwer Academic Publishers.
4. T. Joachims. 1998. Text categorization with support vector machines: learning with many Relevant Features. Proceedings of ECML 98, European Conference on Machine Learning: pp. 137-142. Springer.
5. D.C. Wimalasuriya and D. Dou. 2010. Ontology-Based Information Extraction: an introduction and a survey of current approaches. Journal of Information Science vol. 36(3): pp. 306-323. Sage Publications.
6. A. L. Garrido, O. Gomez, S. Ilarri and E. Mena. 2012. An Experience Developing a Semantic Annotation System in a Media Group. Proceedings of NLDB 12, International Conference on Applications of Natural Language Processing to Information Systems: pp. 333-338. Springer.
7. S. Russell and P. Norvig. 2003. Artificial Intelligence: a modern approach. Prentice-Hall.
8. T.R. Gruber. 1993. A translation approach to portable ontology specifications, Knowledge Acquisition vol. 5(2): pp. 199-220. Academic Press Ltd.
9. G.A. Miller. 1995. WordNet: a lexical database for english. Communications of ACM vol. 38(11): pp. 39-41. ACM.
10. X. Carreras, I. Chao, L. Padró, and M. Padró. 2004. Freeling: an open-source suite of language analyzers. Proceedings of the 4th International Conference on Language Resources and Evaluation: pp. 239-242. European Language Resources Association.
11. P.A. Chirita, S. Costache, W. Nejdl, and S. Handschuh. 2007. P-tag: large scale automatic generation of personalized annotation tags for the web. Proceedings of WWW 07, International Conference on World Wide Web: pp. 845-854. ACM.
12. G. Salton and C. Buckley. 1988. Term weighting approaches in automatic text retrieval. Information Processing and Management, vol. 24(5): pp. 513-523. Pergamon Press, Inc.

13. S. Sekine and E. Ranchhod. 2009. Named entities: recognition, classification and use. John Benjamins.
14. T. Joachims. 2004. SVM-Light version 6.01. Ithaca, NY: Department of Computer Science, Cornell University.
15. E. Leopold and J. Kindermann. 2002. Text categorization with support vector machines. How to Represent Texts in Input Space? Machine Learning vol. 46: pp. 423-444. Kluwer Academic Publishers.
16. M. Fernandez-Lopez and O. Corcho. 2004. Ontological engineering. Springer.
17. G. Schohn and D. Cohn. 2000. Less is more: Active learning with support vector machines. Proceedings of the 17th International Conference on Machine Learning: pp. 839-846. Morgan Kaufmann.
18. N.F. Noy, M. Sintek, S. Decker, M. Crubézy, R.W. Fergerson and M.A. Musen. 2001. Creating semantic web contents with Protégé-2000. IEEE Intelligent Systems vol. 16(2): pp. 60-71. IEEE.
19. D.L. McGuinness, F. van Harmelen. 2004. OWL Web Ontology Language overview. W3C Recommendation. Available at: http://www.w3.org/TR/owl-features/
20. E. Prudhommeaux, A. Seaborne. 2006. SPARQL Query language for RDF. W3C Working Draft. Available at: http://www.w3.org/TR/2006/WD-rdf-sparql-query-20061004/
21. J.J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. 2004. Jena: Implementing the semantic web recommendations. Proceedings of WWW 04, International Conference on World Wide Web: pp. 74-83. ACM.
22. E. Sirin, B. Parsia, B. Cuenca, A. Grau, Y. Kalyanpur. 2007. Pellet: a practical OWL-DL reasoner. Journal of Web Semantics vol. 5(2): pp. 51-53.
23. A. Moschitti and R. Basili. 2004. Complex linguistic features for text classification: a comprehensive study. Proceedings of ECIR 2004, European Conference on Information Retrieval: pp. 181-196. Springer.
24. C. Cortes, V. N. Vapnik. Support-vector networks. Machine Learning vol. 20(3):273-297. Kluwer Academic Publishers.
25. B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. 2003. KIM - semantic annotation platform. Natural Language Engineering vol. 10(3-4): pp. 375-392. Cambridge University Press.
26. A. Maedche, G. Neumann, S. Staab. 2003. Bootstrapping an ontology based information extraction system. Springer.
27. D.W. Embley. 2004. Toward semantic understanding: an approach based on information extraction ontologies. Proceedings of the 15th Australasian Database Conference: pp. 3-12. Australian Computer Society, Inc.
28. L. K. McDowell and M. Cafarella. 2006. Ontology-driven information extraction with OntoSyphon. Proceedings of the 10th International Semantic Web Conference: pp. 428-444. Springer.
29. S. Vogrincic and Z. Bosnic. 2011. Ontology-based multi-label classification of economic articles. Computer Science and Information Systems, vol. 8(1): pp. 101-119. ComSIS Consortium.
30. X. Wu, F. Xie, G. Wu, W. Ding. 2011. Personalized news filtering and summarization on the web. Proceedings of ICTAI 11, International Conference on Tools with Artificial Intelligence: pp. 414-421. IEEE.