Journal of Applied Logic 17 (2016) 48–58

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

Improving ontology-based text classification: An occupational health and security application

Nayat Sanchez-Pi [a,*], Luis Martí [b], Ana Cristina Bicharra Garcia [b]

[a] *Institute of Mathematics and Statistics, Rio de Janeiro State University, Rio de Janeiro, RJ, Brazil*
[b] *Institute of Computing, Fluminense Federal University, Niterói, RJ, Brazil*

ARTICLE INFO

Article history:
Available online 28 September 2015

Keywords:
Text classification
Ontology
Oil and gas industry

ABSTRACT

Information retrieval has been widely studied due to the growing amounts of textual information available electronically. Nowadays organizations and industries are facing the challenge of organizing, analyzing and extracting knowledge from masses of unstructured information for decision making process. The development of automatic methods to produce usable structured information from unstructured text sources is extremely valuable to them. Opposed to the traditional text classification methods that need a set of well-classified trained *corpus* to perform efficient classification; the ontology-based classifier benefits from the domain knowledge and provides more accuracy. In a previous work we proposed and evaluated an ontology-based heuristic algorithm [28] for occupational health control process, particularly, for the case of automatic detection of accidents from unstructured texts. Our extended proposal is more domain dependent because it uses technical terms and contrast the relevance of these technical terms into the text, so the heuristic is more accurate. It divides the problem in subtasks such as: (i) text analysis, (ii) recognition and (iii) classification of failed occupational health control, resolving accidents as text analysis, recognition and classification of failed occupational health control, resolving accidents.

1. Introduction

The need for effective methods of automated Information Retrieval has grown during years because of the amount of unstructured data in natural language form generated in modern organizations [4]. There is a need of performing analysis, decision-making, and knowledge management tasks using this unstructured information.

* Corresponding author.
E-mail address: nayat@ime.uerj.br (N. Sanchez-Pi).

http://dx.doi.org/10.1016/j.jal.2015.09.008
1570-8683/

Nontraditional Information Retrieval strategies are: *text mining* that uncovers previously invisible patterns in existing resources and *text classifications* that is a subfield of data mining which refers generally to the process of deriving high quality of information from a text [7].

Automatic text classification is a task of assigning one or more pre-specified classes to a text, based on its content. Text classification techniques are used in many applications, including e-mail filtering, mail routing, spam filtering, news monitoring, sorting through digitized paper archives, automated indexing of scientific articles, classification of news stories and searching for interesting information on the Web, biomedical applications [8], etc. A good survey of hybrid classifiers systems can be found at [33].

However, it is often the case that a suitable set of well classified trained corpus is not available. Even if one is available, the set may be too small, or a significant portion of the corpus in the training set may not have been classified properly. This creates a serious limitation for the usefulness of the traditional text classification methods.

Our proposal is to use the background knowledge represented by means of an ontology. In the area of computing, the ontological concepts are frequently regarded as classes which are organized into hierarchies. The classes define the types of attributes, or properties common to individual objects within the class. Moreover, classes are interconnected by relationships, indicating their semantic interdependence [30].

In previous work we propose and evaluate an ontology-based heuristic algorithm [28] for occupational health control process, particularly, for the case of automatic detection of accidents from unstructured texts. Our extended proposal is more domain dependent because it uses technical terms and contrast the relevance of these technical terms into the text, so the heuristic is more accurate. It divides the problem in subtasks such as: (i) text analysis, (ii) recognition and (iii) classification of failed occupational health control, resolving accidents.

The rest of this manuscript goes on by describing the theoretical foundations that support it. After that, in Section 3, we describe the elements that are involved in our proposal: (i) the elaboration of the ontology, (ii) the use of a thesaurus as a crawling tool, (iii) the use of the ontology as a classifier, (iv) the compensated classifier using techniques terms. Section 4 proposes an oil and gas industry application scenario: occupational health and security where some comparative experiment are presented. Finally, Section 5 presents some final remarks.

2. Foundations

Due to the ever growing amounts of textual information available electronically, organizations are facing the challenge of organizing, analyzing and extract knowledge from masses of unstructured information for decision making process.

Traditional classification approaches use statistical or machine learning methods to perform the task. Such methods include Naïve Bayes [22], Support Vector Machines [32], Latent Semantic Analysis [9] and many others. A good overview of the traditional text classification methods is presented in [29]. All of these methods require a training set of pre-classified documents that is used for classifier training; later, the classifier can correctly assign categories to other, previously unseen documents.

During the last decades, a large number of machine learning algorithms have been proposed for supervised and unsupervised text categorization. So far, however, existing text categorization systems have typically used the Bag-of-Words model where single words or word stems are used as features for representing document content [26].

However, the work on integrating semantic background knowledge into text categorization is still quite scattered. Early works are: [3,13,6,18]. They use WordNet [11] to improve the text clustering task. WordNet is a network of related words, organized into synonym sets, where each of the sets represents one lexical underlying concept. WordNet has been successfully used both in text categorization and clustering [25].

Text categorization using semantic concepts is a step away from simple word or phrase-based categorization towards a semantics-based classification. Latent Semantic Analysis [21] offers an attractive way to transition from the word-space to the concept-space of related phrases. The extracted concepts can be effectively applied to classical categorization, as presented in [19].

There are also the ontology-based approaches. Ontologies [14] offer knowledge that is organized in a more structural and semantic way. The knowledge represented in a comprehensive ontology can be used to identify concepts in a text. Furthermore, if the concepts in the ontology are organized into hierarchies of higher-level categories, it should be possible to identify the category that best classify the content of the text.

Their use in text categorization is well known. As ontologies provide named entities or terms, and relationship between them, an intermediate categorization step requires matching terms to ontological entities. Afterwards, an ontology can be successfully used for term disambiguating and vocabulary unification, as presented in [2]. Another approach, presented in [24], reinforces co-occurrence of certain pairs of words or entities in the term vector that are related in the ontology. Here, categorization can be based on the recognized and disambiguated named entities, as presented in [15] and [30]. Traditional classification methods can also be enriched with the ontology-based information concerning also the neighborhoods of entities [12,34]. Nevertheless, these ontology-based approaches still lack with regard to accuracy classification.

As explained above, the novelty of our approaches compared to the traditional ones is that our categorization method does not require a training set, which is in contrast to the traditional statistical and probabilistic methods that require a set of pre-classified corpus in order to train the classifier. Our proposals also use a thesaurus for finding non-explicit relations between classification text and ontology terms. This feature widens the domain of the classifier allowing it to respond to complex real-life text and more resilient to ontology incompleteness. Another characteristic is the inclusion of technical terms associated to a weight in the relationship.

3. Proposal

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

In these approaches the ontology effectively becomes the classifier. Consequently, it is no longer needed to carry out a classifier training process with a set of labeled documents, since the ontology should already include all important facts. The proposed approaches require a transformation of the document text into a graph structure, which employs entity matching and relationship identification.

3.1. Ontology-based classification

The Ontology Classifier algorithm [27,28] strategy presented consists in the use an ontology as the key component of our text classification heuristic algorithm. Besides the ontology itself, the algorithm is composed of the following set of modules (see Fig. 1 for details):

1. A lemmatization, stemming and stop-word removing preprocessing. In this work we applied for this task the functionality provided by the Apache Lucene framework [16] and a Portuguese verb infinitive finding module specially developed for this work.
2. A thesaurus for locating words appearing in the text in the ontology. In our case we used a customized version of OpenOffice Brazilian Portuguese thesaurus [1].
3. Set of ontology elements tagged with its corresponding classification label.
4. A thesaurus crawling algorithm that takes care of determining the matching degree of text words with a corresponding ontology term.

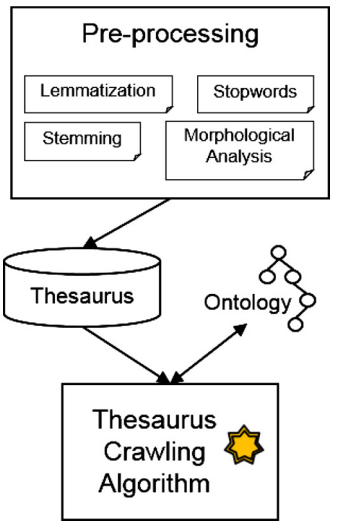

Fig. 1. Process flow diagram.

```
function OntologyClassifier(s, l_max) : M
parameters:
▷ Text as a list of words, s = (s_1, ..., s_n).
▷ Maximum search recursion level, l_max.
returns:
▷ Set of ontology terms (labels) that classify the text, M = {ω_1, ω_2, ..., ω_i, ...}.
begin function
    ŝ ← Preprocess(s).                          ▷ Irrelevant word removal and stemming to roots.
    l_best ← +∞.
    M ← ∅.
    for all s ∈ ŝ do
        Θ ← NearestOntologyTerms(s, 0, l_max).
        for all ⟨θ, l_θ⟩ ∈ Theta do
            if l_best > l_θ then
                M ← {θ}.
                l_best ← l_θ.
            else if l_best = l_θ then
                M ← M ∪ {θ}.
            end if
        end for
    end for
    return ⟨M, l_best⟩.
end function
```

Fig. 2. Pseudo-code description of the algorithm used to compute the levels of similarity between a given word found in text and ontology terms. See Table 1 for a description of the different building-blocks used.

There are some other proposals that also employ mechanisms that rely on ontologies, for example, [10,5], and many more. However, in our case, the use of the thesaurus makes the approach more flexible and capable or handling real-world applications. An ontology describes the application domain in a comprehensive but inflexible way. Natural language is, on the other hand, highly-irregular something rather impossible to grasp by directly using an ontology. The use of the thesaurus along with lemmatization and text processing bridges this gap effectively.

As already mentioned, the classification algorithm proposed in this work relies on the previous ontology, a thesaurus to establish the degree of matching between a given text fragment and some terms of interest that are present in the ontology.

The algorithm is presented as pseudo-code in Fig. 2 as function OntologyClassifier( ). It proceeds by first filtering and rearranging the input sentence in order to render it in a format suitable for processing (Preprocess( ) method in Fig. 3).

Table 1
Summary of utility functions used by the algorithms described in this work.

Function name	Description
Preprocess(s)	Eliminates stop words and finds the roots or infinitive forms.
ConsecutiveWordsCombs(s)	Generates a set that contains all possible consecutive words combinations in s.
OntologyContains(s)	Boolean function that indicates when term s is present in the ontology.
OntologyTechnicalSynonyms(s)	Returns a set of ontology terms that contain s in their technical synonym list.

function NearestOntologyTerms(s, l_{curr}, l_{max}) : Θ
parameters:
▷ A search term, s.
▷ Current recursion level, l_{curr}.
▷ Maximum search recursion level, l_{max}.
returns:
▷ Set of ontology terms (labels) that classify the text with their corresponding levels of similarity, $\Theta = \{\langle \theta_1, l_1 \rangle, \ldots, \langle \theta_i, l_i \rangle, \ldots\}$.
begin function
  if $l_{\text{curr}} = l_{\text{max}}$ **then**
    return $\Theta = \varnothing$.
  end if
  if OntologyContains(s) **then** ▷ The term s is present in the ontology as-is.
    return $\Theta = \{\langle s, l_{\text{curr}} \rangle\}$.
  end if
  $l_{\text{best}} \leftarrow +\infty$.
  $\Theta \leftarrow \varnothing$.
  for all $s^* \in$ Thesaurus(s) **do** ▷ Recursively crawl the thesaurus.
    $\Theta^* \leftarrow$ NearestOntologyTerms(s^*).
    $\Theta \leftarrow \Theta \cup \Theta^*$
  end for
  return Θ.
end function

Fig. 3. Pseudo-code description of the algorithm used to compute the levels of similarity between a given word found in text and ontology terms.

Having the filtered text represented as a set of words, the algorithm proceeds to identify which terms of the ontology are most closely related to that set. It carries that out by invoking for each word the function NearestOntologyTerms( ). This function—which is described in Fig. 3—returns the set of ontology terms that are related with a given word by recursively traversing a thesaurus up to a given number of levels. If a connection between a word and a term is established that term is included, along with its level of similarity in the set of related terms Θ. The level of similarity is defined as the number of jumps needed to get from to word to the term using the thesaurus. A lower level implies higher similarity.

The result of the classification is one or more ontology terms that are most closely related to the text, or, posed in other words, the terms with minimal level of similarity. It should be borne in mind that the two functions presented here have been simplified for didactic reasons, and in practice some a harder to read but more efficient option is used.

3.2. Improving classification

The previous classification algorithm has been applied with success in real-life problems. This experience is also usefull in order to point out issues that should be addressed in order to yield further improved results. These issues can be summarized as:

1. *Focus on multi-word thesaurus.* The use of single-word thesaurus limits the expression capacity and reach of the algorithm.
2. *Use technical terms synonyms.* The ontology is meant to describe the application context in detail and accuracy. Nevertheless, there are a number of concepts that can be referred by different names.

Each of these names might have a different degree of relationship with the one used by the ontology. Classification algorithms should be capable of dealing with such term synonyms.

3. *Relevance of a term within a text.* The relation of the length of the ontology term with regard to the length of the whole text should be taken into account. It should not have the same impact a relatively short term in shorter or longer texts. Similarly, longer terms should have higher weight in the classification, as it can be hypothesized that longer terms tend to describe an artifact in higher detail [17].

A number of improvements can be made to OntologyClassifier in order to deal with and solve the shortcomings derived form the previous issues.

Regarding the first issue, the original thesaurus can be enriched by using a problem vocabulary. This vocabulary can be—and was—obtained by conferencing with the experts of the field. As this can be a tedious process we resorted to a frequent n-grams generation algorithm [23] to assist in it. The set of frequent n-grams extracted form all available classification text contains words that repeatedly appeared in consecutive form. An expert can filter relevant n-grams and create associations between them. It should be noted that the n-gram approach is also of help when validating the degree at which the ontology respond to the actual text to be classified.

The generation of term synonyms employed a similar approach. Each concept in the ontology was annotated with a list of technical synonyms that would be taken into account at the same level as the original concept. Experts with the help of the frequent n-gram list populated the list of concept synonyms.

The third issue called for integrative relevance measure that combined the impact of current search text with regard to the total text and the level of similarity of the search text with its corresponding ontology terms. The length-compensated similarity measure is proposed with that requirement in mind. This measure combines the aforementioned quantities as the ratio,

$$c_{\text{comp}} = \frac{|s_{\text{search}}|}{(l+1)\,|s_{\text{text}}|}, \tag{1}$$

where s_{search} is the text being used for search, l is the level of similarity yielded by the ontology and thesaurus crawling algorithm, s_{text} is all the text being classified and the function $|\cdot|$ measures the length of the text passed as argument.

The a novel classification algorithm TermRelevanceOntologyClassifier is proposed by combining the previously described modifications. It is presented in summarized form in Fig. 4. The new algorithm maintains an structural similarity with the previous one. The main difference lies in the fact that it makes use of all possible consecutive word combinations. Subsequently, the NearestOntologyTermsOrSynonyms method—which is outlined in Fig. 5—is applied to each word combination. The length-compensated measure is used to prime longer terms with regard to others.

Although the new algorithm has the potential of yielded better results it should be pointed out that by exploring all possible word combinations the computational cost of the algorithm skyrockets at an exponential rate. However, in the application context of the algorithm this rises on computation time. It should also be pointed out that although the algorithms are presented in a recursive form, for didactic reasons, they were implemented in a high-performance, parallel form with a number of speed-ups meant to avoid excessive search as results were obtained.

4. Case study: occupational health and security in oil industry

Occupational health and security (OHS) issues are priority matter for the offshore oil and gas industry. This industry is frequently in the news. Much of the time it is because of changes in prices of oil and gas. Other—less frequent but perhaps more important—subject of media attention is when disasters strike, as

function TermRelevanceOntologyClassifier(s, $l_{\max}$) : $\mathcal{M}$
parameters:
▷ Text string, s.
▷ Maximum search recursion level, $l_{\max}$.
returns:
▷ Set of similar ontology terms (labels), $\mathcal{M}$.
begin function
  $\hat{s} \leftarrow$ Preprocess(s). ▷ Irrelevant word removal and stemming to roots.
  $\mathcal{C} \leftarrow$ ConsecutiveWordsCombs($\hat{s}$). ▷ Set of all consecutive words combinations.
  $\mathcal{M} \leftarrow \varnothing$. ▷ Start with empty results.
  $c_{\text{best}} \leftarrow 0$.
  for all $\gamma \in \mathcal{C}$ **do**
    $\Theta \leftarrow$ NearestOntologyTermsOrSynonyms($\gamma, 0, l_{\max}$).
    for all $\langle \theta, l_\theta \rangle \in \Theta$ **do**
      Compute length-compensated similarity (1) as,

$$c = \frac{|\gamma|}{(l_\theta + 1)\,|\hat{s}|}.$$

      if $c_{\text{best}} < c$ **then** ▷ Keep most similar ontology terms.
        $\mathcal{M} \leftarrow \{\gamma\}$.
        $c_{\text{best}} \leftarrow c$.
      else if $c_{\text{best}} = c$ **then**
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{\theta\}$.
      end if
    end for
  end for
  return $\mathcal{M}$.
end function

Fig. 4. Pseudocode of the ontology-based classifier that uses term relevance to weight ontology classification terms and the technical synonyms list. See Table 1 for a description of the different building-blocks used.

function NearestOntologyTermsOrSynonyms(s, l_{curr}, $l_{\max}$) : Θ
parameters:
▷ A search term, s.
▷ Current recursion level, l_{curr}.
▷ Maximum search recursion level, $l_{\max}$.
returns:
▷ Set of ontology terms (labels) that classify the text with their corresponding levels of similarity, $\Theta = \{\langle \theta_1, l_1 \rangle, \ldots, \langle \theta_i, l_i \rangle, \ldots\}$.
begin function
  if $l_{\text{curr}} = l_{\max}$ **then**
    return $\Theta = \varnothing$.
  end if
  if OntologyContains(s) **then** ▷ The term s is present in the ontology as-is.
    return $\Theta = \{\langle s, l_{\text{curr}} \rangle\}$.
  end if
  $\mathcal{S} \leftarrow$ OntologyTechnicalSynonyms(s).
  if $\mathcal{S} \neq \varnothing$ **then** ▷ s appears as a technical synonym of an element of the ontology.
    return $\Theta \leftarrow \{\langle s_{\text{sym}}, l_{\text{curr}} \rangle, \forall s_{\text{sym}} \in \mathcal{S}\}$.
  end if
  $l_{\text{best}} \leftarrow +\infty$.
  $\Theta \leftarrow \varnothing$.
  for all $s^* \in$ Thesaurus(s) **do** ▷ Recursively crawl the thesaurus.
    $\Theta^* \leftarrow$ NearestOntologyTermsOrSynonyms(s^*).
    $\Theta \leftarrow \Theta \cup \Theta^*$
  end for
  return Θ.
end function

Fig. 5. An improved term similarity algorithm that takes into account technical terms synonyms.

is the case of offshore oil drilling platform explosions, spills or fires. These incidents have a high impact on lives, environment and public opinion regarding this sector. That is why a correct handling of OHS is a determining factor in this industry long-term success.

There is an important effort of oil and gas industry to reduce the number of accidents and incidents. There are standards to identify and record workplace accidents and incidents to provide guiding means on

Table 2
Classification performance measures yielded by the three algorithms being compared.

	SVM classifier	Ontology classifier	Term relevance OC
Accuracy	0.5420	0.6543	0.9432
Precision	0.6401	0.6396	0.9620
Recall	0.6140	0.6735	0.9643
F-score	0.6831	0.7743	0.8786
Specificity	0.7673	0.8643	0.9524

prevention efforts, indicating specific failures or reference, means of correction of conditions or circumstances that culminated in accident. Besides, oil and gas industry is increasingly concerned with achieving and demonstrating good performance of occupational health and safety (OHS), through the control of its risks, consistent with its policy and objectives.

4.1. An occupational health and safety ontology

As part of this work, we devised a domain ontology for Occupational Health and Security (OHS) in oil and gas application context. This ontology was elaborated after interviewing field experts, an extensive reviewing of related literature and the analysis of the existing data sources.

Here we obtained the inferences that describe the dynamic side and finally we group the inferences sequentially to form tasks. The principal concept of the ontology is *anomaly* which is an undesirable event or situation which results or may result in damage or faults that affect people, the environment, equity (own or third party), the image of the multinational petroleum system, products or production processes. This concept includes accidents, illnesses, incidents, deviations and non-conformances. We direct the interested reader to our previous work [28] for further details.

4.2. Comparative experiments

Experiments are necessary in order to compare the approaches discussed above and empirically contrast the performance and improvements introduced by the novel Term Relevance Ontology Classifier with regard to the regular Ontology Classifier.

The algorithms presented in this paper make use of the domain ontology described in the previous section. Because of that, the tests that can be carried out are limited to the OHS domain, and, therefore, makes impossible straightforward assessment of the algorithms in other well-known benchmark problems. In any case, although bound to the application domain, experiments are important.

We prepared an experiment dataset containing the descriptive fields of 500 anomalies with that purpose. We labeled these anomalies by hand using existing ontology terms and applied the previous two algorithms to verify at what degree the text was correctly labeled.

In order to provide grounds for comparison we contrast the algorithms performance with the one yielded by a set of support vector machines [32]. Each SVM is trained as binary classifiers for each anomaly label using a methodology equivalent to the one proposed by [20]. A 10-fold cross-validation was performed using a 70/30 ratio for training and testing. The results shown are the most voted one among the 10 classifiers.

The non-stochastic nature of the algorithms discussed simplifies the experimentation at great length as they are no sensible to training set sizes, ordering or initialization.

Table 2 summarizes the classification performance indicators [31] yielded three methods being tested. In particular, it shows the *accuracy*, which measures the overall effectiveness of a classifier; the *precision*, that is a measure of class agreement of the data labels with the positive labels given by the classifier; *recall*, that shows the effectiveness of a classifier to identify positive labels; *F-score*, that relates data's positive

labels and those given by a classifier, and; *specificity*, that assesses the classifier capacity to identify negative labels.

The results can be interpreted with a twofold analysis. First, it is noticeable that ontology-based methods outperform the pure machine-learning approach. Even if this was expected, this results speaks in favor of the application of such approaches, or similar ones to this class of problems. Of course, ontology-based classification have the disadvantage derived from the requirement of having an extensive *a priori* knowledge of the problem. On the other hand, there are cases—as the problem previously discussed—when such knowledge is available, while preparing a proper annotated dataset to be use as part of a machine learning approach implied an inviable temporal and manpower cost.

The second important results goes beyond the discussion comparing machine learning and ontology-based classifiers. The experimental evidence points out that the modifications introduced with the Term Relevance Ontology Classifier yield a substantial improvement with regard to all performance metrics.

5. Final remarks

In this paper we presented two novel text classification methods based on leveraging the existing knowledge represented in domain ontology. We have focused our approach on a real-life high-relevance problem: the health, safety and environment issues in the oil and gas industry.

The novelty of these approaches is that they is not depend on the existence of a training set, as it relies solely on the ontology entities, their relationships, and the taxonomy of categories represented by them. It might be argued that the synthesis of such ontology is comparable at some degree with the preparation of an annotated training set. However, when analyzing this issue at a deeper level it may be realized that an ontology-based solution is better mainly because an ontology can be easily contrasted and verified, both by formal means and by members of the research team. Therefore, this approach is less prone to bias, inconsistency and prejudice. Similarly, the resulting ontology is a relevant asset on its own right.

The first approach, termed Ontology Classifier, incorporates a thesaurus for overcoming the possible narrow classification domain imposed by the limited set of terms that are present in the ontology. Hence, the similarity of a given search term with the number of jumps that are necessary to reach an ontology element starting from the text. This feature makes the method more flexible and resilient to real-life texts that are hardly written in a homogeneous or exact form.

The second proposal, the Term Relevance Ontology Classifier, improves the first one by adding the use of a technical synonyms list attached to ontology elements. These lists are generated in a semi-automatic way using an n-gram extraction algorithm. It also incorporated a new similarity criterion that balances the level of similarity used in the previous case with the relevance of the given search term with regard to the overall text.

The experimental comparison carried out showed that both algorithms yielded a better performance when compared to an state-of-the-art machine learning approach. Similarly, these tests also showed the substantial performance improvements obtained with the modification put forward by the second approach.

It must be said that this paper presents a set of results that is susceptible of being improved. In particular, we are interested on using the ontology to provide a more granular classification.

Finally, this paper does not imply, nor should not be taken as, an argument against machine learning. Machine learning approaches are capable of extracting information from large masses of data, while, on the other hand, these proposal focus on localized, focused knowledge. In this regard we are currently developing a hybrid approach that could be able to exploit the benefits of both methodologies.

Acknowledgement

This work was partially funded by CNPq BJT Project 407851/2012-7.

References

[1] Apache OpenOffice.org, DicSin: Dicionário de sinônimos Português/Brasil, http://extensions.openoffice.org/en/project/DicSin-Brasil, 2013.

[2] S. Bloehdorn, A. Hotho, Text classification by boosting weak learners based on terms and concepts, in: Fourth IEEE International Conference on Data Mining, ICDM'04, IEEE, 2004, pp. 331–334.

[3] R.C. Bodner, F. Song, Knowledge-Based Approaches to Query Expansion in Information Retrieval, Springer, 1996.

[4] M.L. Borrajo, B. Baruque, E. Corchado, J. Bajo, J.M. Corchado, Hybrid neural intelligent system to predict business failure in small-to-medium-size enterprises, Int. J. Neural Syst. 21 (04) (2011) 277–296.

[5] F. Camous, S. Blott, A.F. Smeaton, Ontology-based MEDLINE document classification, in: Bioinformatics Research and Development, Springer, 2007, pp. 439–452.

[6] K. Dave, S. Lawrence, D.M. Pennock, Mining the peanut gallery: opinion extraction and semantic classification of product reviews, in: Proceedings of the 12th International Conference on World Wide Web, ACM, 2003, pp. 519–528.

[7] F. De la Prieta, A.B. Gil, S. Rodríguez, J.B. Pérez, J.A.G. Coria, J.M. Corchado, An enhanced approach to retrieve learning resources over the cloud, in: The 2nd International Workshop on Learning Technology for Education in Cloud, Springer, 2014, pp. 193–203.

[8] J.F. De Paz, J. Bajo, V.F. López, J.M. Corchado, Biomedic organizations: an intelligent dynamic architecture for KDD, Inf. Sci. 224 (2013) 49–61.

[9] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (6) (1990) 391–407, http://dx.doi.org/10.1002/(SICI)1097-4571(199009).

[10] J. Fang, L. Guo, X. Wang, N. Yang, Ontology-based automatic classification and ranking for Web documents, in: Fourth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 3, FSKD'2007, IEEE, 2007, pp. 627–631.

[11] C. Fellbaum, WordNet, Wiley Online Library, 1999.

[12] E. Gabrilovich, S. Markovitch, Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge, in: 21st National Conference on Artificial Intelligence, vol. 6, AAAI-06, 2006, pp. 1301–1306.

[13] S.J. Green, Building hypertext links by computing semantic similarity, IEEE Trans. Knowl. Data Eng. 11 (5) (1999) 713–730.

[14] T.R. Gruber, A translation approach to portable ontology specifications, Knowl. Acquis. 5 (2) (1993) 199–220.

[15] B. Hammond, A. Sheth, K. Kochut, A modular document enhancement platform for semantic applications over heterogeneous content, in: Real World Semantic Web Applications, vol. 92, 2002, p. 29.

[16] E. Hatcher, O. Gospodnetic, M. McCandless, Lucene in Action, Manning Publications, Greenwich, 2004.

[17] T. Hirsimäki, J. Pylkkonen, M. Kurimo, Importance of high-order n-gram models in morph-based speech recognition, IEEE Trans. Audio Speech Lang. Process. 17 (4) (2009) 724–732, http://dx.doi.org/10.1109/TASL.2008.2012323.

[18] A. Hotho, S. Staab, G. Stumme, Ontologies improve text document clustering, in: Third IEEE International Conference on Data Mining, ICDM 2003, IEEE, 2003, pp. 541–544.

[19] Y. Huang, Support vector machines for text categorization based on latent semantic indexing, Tech. rep., Electrical and Computer Engineering Department, the Johns Hopkins University, 2003.

[20] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: C. Nédellec, C. Rouveirol (Eds.), Machine Learning: ECML-98, in: Lecture Notes in Computer Science, vol. 1398, Springer, Berlin/Heidelberg, 1998, pp. 137–142.

[21] T.K. Landauer, P.W. Foltz, D. Laham, An introduction to latent semantic analysis, Discourse Process. 25 (2–3) (1998) 259–284.

[22] D.D. Lewis, Naive (Bayes) at forty: the independence assumption in information retrieval, in: C. Nédellec, C. Rouveirol (Eds.), Machine Learning, ECML-98, in: Lecture Notes in Computer Science, vol. 1398, Springer, 1998, pp. 4–15.

[23] H. Masataki, Y. Sgisaka, Variable-order n-gram generation by word-class splitting and consecutive word grouping, in: 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, ICASSP'96, 1996, pp. 188–191.

[24] M. Nagarajan, A. Sheth, M. Aguilera, K. Keeton, A. Merchant, M. Uysal, Altering document term vectors for classification: ontologies as expectations of co-occurrence, in: Proceedings of the 16th International Conference on World Wide Web, ACM, 2007, pp. 1225–1226.

[25] P. Rosso, E. Ferretti, D. Jiménez, V. Vidal, Text categorization and information retrieval using wordnet senses, in: Proceedings of the Second International Conference of the Global WordNet Association, 2004, pp. 299–304.

[26] G. Salton, Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, 1989.

[27] N. Sanchez-Pi, L. Martí, A.C. Bicharra Garcia, Information extraction techniques for health, safety and environment applications in oil industry, in: International Conference Intelligent Systems and Agents, IADIS, 2013, pp. 115–117.

[28] N. Sanchez-Pi, L. Martí, A.C. Bicharra Garcia, Text classification techniques in oil industry applications, in: A. Herrero, B. Baruque, F. Klett, A. Abraham, V. Snášel, A.C. Carvalho, P. García Bringas, I. Zelinka, H. Quintián, E. Corchado (Eds.), International Joint Conference SOCO'13–CISIS'13–ICEUTE'13, in: Advances in Intelligent Systems and Computing, vol. 239, Springer International Publishing, 2014, pp. 211–220.

[29] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (1) (2002) 1–47.

[30] A. Sheth, C. Bertram, D. Avant, B. Hammond, K. Kochut, Y. Warke, Managing semantic content for the web, IEEE Internet Comput. 6 (4) (2002) 80–87.

[31] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (4) (2009) 427–437, http://dx.doi.org/10.1016/j.ipm.2009.03.002, http://www.sciencedirect.com/science/article/pii/S0306457309000259.

[32] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 2000.
[33] M. Woźniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (2014) 3–17.
[34] S.-H. Wu, T.-H. Tsai, W.-L. Hsu, Text categorization using automatically acquired domain ontology, in: Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, vol. 11, Association for Computational Linguistics, 2003, pp. 138–145.