Open Access

Open Journal of Semantic Web (OJSW)
Volume 3, Issue 1, 2016
http://www.ronpub.com/ojsw
ISSN 2199-336X

Hierarchical Multi-Label Classification Using Web Reasoning for Large Datasets

Rafael Peixoto[A,B], Thomas Hassan[B], Christophe Cruz[B], Aurélie Bertaux[B], Nuno Silva[A]

[A] ISEP, Polytechnic of Porto, Rua Dr. António Bernardino de Almeida, 431, Porto, Portugal,
{rafpp, nps}@isep.ipp.pt
[B] LE2I UMR6306, Univ. Bourgogne Franche-Comté, Bâtiment I3M salle 111, 64B rue de Sully, Dijon, France,
{thomas.hassan, christophe.cruz,aurelie.bertaux }@u-bourgogne.fr

Abstract

Extracting valuable data among large volumes of data is one of the main challenges in Big Data. In this paper, a Hierarchical Multi-Label Classification process called Semantic HMC is presented. This process aims to extract valuable data from very large data sources, by automatically learning a label hierarchy and classifying data items.The Semantic HMC process is composed of five scalable steps, namely Indexation, Vectorization, Hierarchization, Resolution and Realization. The first three steps construct automatically a label hierarchy from statistical analysis of data. This paper focuses on the last two steps which perform item classification according to the label hierarchy. The process is implemented as a scalable and distributed application, and deployed on a Big Data platform. A quality evaluation is described, which compares the approach with multi-label classification algorithms from the state of the art dedicated to the same goal. The Semantic HMC approach outperforms state of the art approaches in some areas.

Type of Paper and Keywords

Regular research paper: *Multi-label classification, hierarchical classification, Big Data, ontology, machine learning, web reasoning, large dataset*

1 Introduction

The item analysis process requires proper techniques for analysis and representation. In the context of Big Data, this task is even more challenging due to Big Data's characteristics. An increasing number of V's has been used to characterize Big Data [5, 14]: Volume, Velocity, Variety and Value. Volume concerns the large amount of data that is generated and stored through the years by social media, sensor data, etc. [5]. Velocity concerns both the production and the process to meet a demand because Big Data is not only a huge volume of data but it must be processed quickly as new data is generated over time. Variety relates to the various types of data composing the Big Data. These types include semi-structured and unstructured data representing 90% of his content [30] such as audio, video and text. Value means how valuable the information to a Big Data consumer is. Value is the most important feature of Big Data and its "raison d'être", because the user expects to make profit out of valuable data.

Big Data analysis can be deemed as the analysis technique for a special kind of data. Therefore, many traditional data analysis methods used in Data Mining (algorithms for classification, clustering, regression, among others) may still be utilized for Big Data Analysis [5]. Werner et al. [41] propose a method to semantically

enrich an ontology used to describe the domain and classify the news articles. This ontology aims to reduce the gap between the expert's perspective and the classification rules representation. To enrich the ontology and classify the documents they use an out-of-the-box Description Logics (DL) Web Reasoner like Pellet [28], FaCT++ [32], or Hermit [27]. Most of these reasoners are sound and complete to high expressiveness, such as OWL2 SROIQ (D) expressiveness, but on the other hand they do not scale [41]. They are good enough for a proof of concept but when the number of documents, words and taxonomies increases, these reasoners cannot handle a large amount of data. Our goal is to extend the work in [41] and to exploit value by analyzing Big Data using a Semantic Hierarchical Multi-Label Classification process (Semantic HMC) [13]. Hierarchical Multi-Label Classification (HMC) is the combination of Multi-Label classification and Hierarchical classification [3].

The Semantic HMC is based on an unsupervised ontology learning process using scalable Machine-Learning techniques and Rule-based reasoning. The process is unsupervised such that no previously classified examples or rules to relate the data items with the labels exist. The ontology-described knowledge base (Abox+Tbox) used to represent the knowledge in the classification system is automatically learned from huge volumes of data through highly scalable Machine Learning techniques and Big Data Technologies. First, the hierarchy of labels is automatically obtained and used as the first input for the ontology construction [10]. Then, for each classification label a set of rules is created to relate the data items to the labels. Finally, the learned ontology is populated with the data items. Therefore, the Semantic HMC proposes five individually scalable steps (Fig. 1) to reach the aims of Big Data analytics [13, 25] :

- **Indexation** extracts terms from data items and creates an index of data items. In the example of indexing text documents, the extracted terms are relevant words to describe an item. The term extraction includes treatments such as spelling correction as well as stop-word and synonym detection and composed terms calculation by collocation. A collocation is a sequence of words (n-grams), which co-occur more often than it would be expected by chance. An association measure algorithm evaluates whether the co-occurrence is purely by chance or statistically significant. The inverted index allows efficient retrieval of documents by term.
- **Vectorization** calculates the term-frequency vectors of the indexed items by calculating the term frequency of each term in the collection of documents (i.e. Document Frequency and TF-IDF). Further, a term co-occurrence frequency matrix is created to represent the co-occurrence of any pair of terms in a document. The calculated matrix is exploited in the hierarchization step to create the subsumption relations and in the resolution step to create the classification rules. The vectors are used lately in the realization step to describe the data items with relevant terms.
- **Hierarchization** creates the label taxonomy (i.e. subsumption hierarchy) by exploiting the co-occurrence matrix. As it is an unsupervised process where no labels are previously defined, the most relevant terms are designated as labels. To calculate the term relevance a ranking method based on information retrieval is used. A subsumption algorithm is hence used to automatically calculate the hierarchical relations between labels. The result of this step is a subsumption hierarchy of labels, and more specifically a Directed Acyclic Graph (DAG).
- **Resolution** describes the taxonomy concepts (labels) using their related terms and creates the classification rules used to classify data items with labels, i.e. it establishes the conditions for an $item1$ to be classified in $label1$. The created classification rules define the necessary and sufficient terms for an item to be classified with a label. The rules are serialized in a horn clause language. The main interest in using horn clause rules instead of translating the rules into logical constraints of an ontology captured in Description Logic is to reduce the reasoning effort, thus improving the scalability and performance of the system.
- **Realization** The realization step populates the ontology and performs the multi-label hierarchical classification of the items. For that, the ontology is first populated with items and the most relevant terms to describe each item in an assertion level (Abox). A rule-based inference engine is then used to infer the most specific labels as well as all the broader concepts for each item according to the label hierarchy. This leads to a multi-label classification of the items based in a hierarchical structure of the labels (Hierarchical Multi-label Classification).

The first three steps learn the label hierarchy from unstructured data as described in [25]. As a follow up, [24] focuses on the last two steps of the Semantic HMC process. It proposes a new process to hierarchically multi-classify items from huge sets of unstructured texts

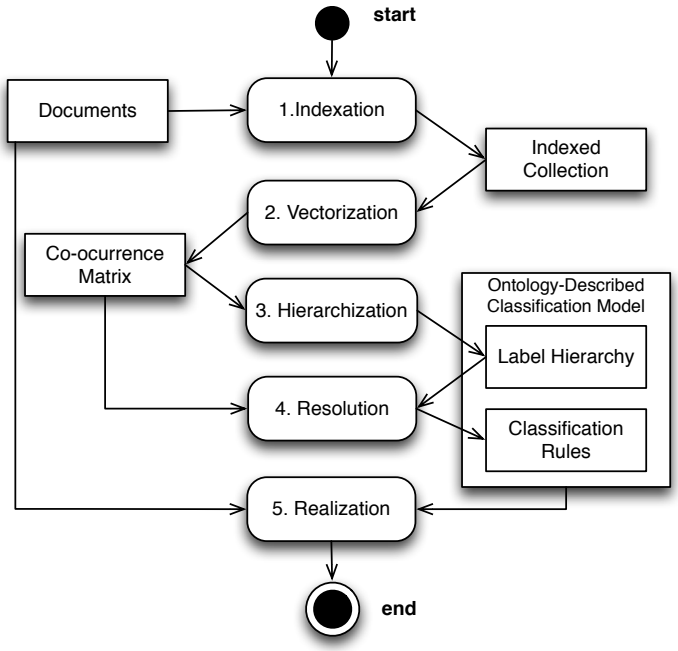


Figure 1: Semantic HMC process steps

using DL ontologies and Rule-based reasoning. The process is implemented using scalable approaches that distribute the process to several machines in order to reach high performance and scalability required by Big Data.

This paper is an extention of the work presented in [24] and provides extended experiments with quality evaluation and comparison with some multi-label classification algorithms from the state of the art. The rest of the paper covers five sections. The second section presents background and related work. The third section describes the classification process. The forth section describes the process implementation in a scalable and distributed platform to process Big Data. The fifth section discusses the results. Finally, the last section draws conclusions and suggests further research.

2 Related Work

In this section, we introduce some background and discuss the current related work about automatic hierarchical multi-label classification for unstructured text using DL ontologies and reasoning. The following subsections discuss the related work of ontologies and web reasoning in classification context.

2.1 Multi-label Classification

The Multi-Label Classification (MLC) method classifies an item with one or more labels from a subset of labels Y where $Y \subseteq L$ and L is the complete set of labels.

The Binary Relevance (BR) [33] addresses the multi-label learning problem by learning one classifier for each label. Each binary classifier determines the relevance of its label for the current item, resulting in a set of relevant labels. The probability of relevance can be used to rank the labels outputted by each classifier.

Classifier chaining (CC) method [26] uses binary classifiers as in the BR approach. Q different classifiers are linked along a chain, where the $i-th$ classifier deals with the binary relevance problem associated with label $\lambda_i \in L$ where $(1 \leq i \leq Q)$. The feature space of each link between classifiers in the chain is extended with the 0/1 label associations of all previous links. Prediction and ranking of the relevant labels are determined in the same way as the binary relevance method.

TNBCC[1] and Path-BCC are two approaches derived from Chain Classifiers [29]. For each label Li in the chain, a naïve Bayes classifier is defined. In Path-BCC, all the subsuming nodes in the chain from the root label to the label Li are considered as attributes of the label

[1] TNBCC and Path-BCC are the real names of the approaches and not acronyms. However, TNBCC can be described as Tree Naïve Bayesian Chain Classifier and Path-BCC is described as Path-Bayesian Chain Classifier.

Li, while TNBCC only includes the parent label, hence Path-BCC uses more contextual information.

The Hierarchy Of Multi-label Classifiers (HOMER) [34] is an algorithm for effective and efficient multi-label learning, designed for large multi-label datasets. HOMER constructs a hierarchy of multi-label classifiers, where each classifier is dedicated to a small set of labels. This allows the process to scale by evenly distributing the items among the classifiers.

The Multi-Label K-Nearest Neighbors (ML-kNN) [43] is an extension of the k-nearest neighbours (kNN) algorithm. First, for each $item_i$, its k-nearest neighbours in the training set are identified, based on its features. Then, a posteriori principle is used to determine the label set for the $item_i$, based on statistical information extracted from the label sets of its neighbours, i.e., the number of neighbouring examples belonging to each possible label.

The Random Forest of Predictive Clustering Trees (RF-PCT) and the Random Forest of ML-C4.5 (RFML-C4.5) [18] are ensemble methods that respectively use PCTs and ML-C4.5 trees, as base classifiers. Several base classifiers are used to determine the relevance of labels. The predictions made by the classifiers are then combined using voting scheme such as majority or probability distribution vote.

The Latent Dirichlet Allocation (LDA) is a generative probabilistic model for discovering the underlying structure of discrete data. The Collapsed Gibbs Sampling (CGS) algorithm [12] approximates the LDA parameters of interest through iterative sampling in a Markov Chain Monte Carlo (MCMC) procedure. Unlike CGS, the final predictions made in the CGS_p[23] approach are based on a Collapsed Variational Bayesian inference update procedure, i.e. CVB0 [1].

2.2 Ontologies in Classification Context

The ontologies are recurrently used in classification systems to describe the classification knowledge (labels, items, classification rules) and to improve the classification process.

Ontologies are a good solution for intelligent computer systems, and operate close to the human concept level bridging the gap between the human requirements and the computational requirements [22]. Galinina et al. [11] used two ontologies to represent a classification system: (1) a domain ontology that is independent of any classification method and (2) a method ontology devoted to decision tree classification. Beyond the domain description, ontologies can be used to improve the classification process. Elberrichi et al. [7] present a two-step method for improving classification of medical documents using domain ontologies (MeSH - Medical Subject Headings). Their results prove that document classification in a particular area supported by ontology of its domain increases the classification accuracy. Johnson et al. [16] propose an iterative and interactive (between AI methods and domain experts) approach to achieve prediction and description ("which are usually hard to fulfill"), considering domain expert knowledge and feedback. Vogrincic et al. [40] are concerned with automatically creating an ontology from text documents without any prior knowledge about their content.

2.3 Web Reasoning in Classification Context

Reasoning is used at the ontology development or maintenance time as well as at the time ontologies are used for solving application problems [20]. Web reasoning can be used to improve the classification process.

In [8] authors present a document classification method that uses ontology reasoning and similarity measures to classify the documents. In [2] authors introduce a generic, automatic classification method that uses Semantic Web technologies for defining the classification requirements, performing the classification and representing the results. This method allows data elements from diverse sources and of different formats and types to be classified using an universal classification scheme. The proposed generic classifier is based on an ontology, which gives a description of the entities that need to be discovered, the classes to which these entities will be mapped, and information on how they can be discovered. In [41], the authors propose a method to semantically enrich the ontology used to hierarchically describe the domain and to process the classification of news using the hierarchy of terms. This ontology aims at reducing the gap between the expert's perspective and the classification rules representation. To enrich the ontology and classify the documents a DL Web Reasoner like Pellet [28], FaCT++ [32], or Hermit [27] is used.

2.4 Discussion

Most work in the literature focus on describing or improving the classification processes using ontologies, but do not take advantage of the reasoning capabilities of web reasoning to automatically multi-classify the items.

In [41] authors use out-of-the-box reasoning to classify economical documents, but their scalability is limited and cannot be used in large datasets such as required in Big Data context. As Semantic Web is growing, new high-performance Web Scale Reasoning methods have been proposed [35]. Rule-based reasoning approaches allow the parallelization and distribution

of work by large clusters of inexpensive machines by programming models for processing and generating large data sets as MapReduce [6]. Web Scale Reasoners [35], instead of using traditional DL approaches like Tableau [28, 32], Resolution [21] or Hypertableau [27], use entailment rules for reasoning over ontologies. The Web-Scale Reasoners based on the Map-Reduce programming model like WebPie [36] outperform all other published approaches in an inference test over 100 billion triples [37]. the recent implementations of Web-Scale Reasoners as WebPie are limited to low expressive ontologies as OWL-Horst fragment [31] due to the complexity of implementation and performance at web scale. In [42] authors describe a kind of semantic web rule execution mechanism using MapReduce which can be used with OWL-Horst and with SWRL rules.

To the extent of our knowledge, a classification process to automatically classify text documents in Big Data context by taking advantage of ontologies and rule-based reasoning to perform the classification is novel.

3 HIERARCHICHAL MULTI-LABEL CLASSIFICATION

In this section the last two steps (Resolution and Realization) of the hierarchical multi-label classification process are described in detail. In [25], the authors describe in detail the first three steps (Indexation, Vectorization and Hierarchization) of the classification process. The ontology-described label hierarchy is automatically learned from huge volumes of unstructured text documents using Big Data technologies. Beyond learning the label hierarchy, this paper aims to learn a classification model based on a DL ontology presented in Table 1. Establishing $isClassified$ relationships between $Item$ and $Label$, as described in the ontology, considering scalability, is the final goal of this paper.

The following subsections describe (i) the process background, (ii) how the rules used to classify the items are created and (iii) the item classification using Rule-based Web Reasoning.

3.1 Resolution

The resolution step creates the ontology rules used to relate the labels and the data items, i.e. it establishes the conditions for an $item_i$ to be classified as $label_j$. The rules will define the necessary and sufficient terms of an item $item_i$ be classified as $label_j$. The process of rule creation uses thresholds as proposed in [41] to select the necessary and sufficient terms. The main difference of our method compared to [41] is that instead of translating the rules into logical constraints of an ontology captured in Description Logic, these rules are translated into horn clause rules (i.e. W3C Rule Interchange Format (RIF) or Semantic Web Rule Language (SWRL) [15]). The main interest in using horn clause rules is to reduce the reasoning effort, thus improving the scalability and performance of the system. The aim is to use more, but simpler horn clause rules that will be applied to the ontology in order to classify items. We use the SWRL language for the examples and implementation because it is the only language supported across all the technologies we use for evaluation.

In Vectorization step, a term co-occurrence frequency matrix $cfm(term_i, term_j)$ is created to represent the co-occurrence of any pair of terms in the collection of items C. Let $P(term_j|term_i)$ be the conditional proportion (number) of the items from collection C common to $term_i$ and $term_j$, in respect to the number of items in $term_j$ such that:

$$P_C(term_i|term_j) = \frac{cfm(term_i, term_j)}{cfm(term_j, term_j)} \quad (1)$$

Two thresholds are defined:

- Alpha threshold (α) such that $\alpha < P_C(term_i|term_j)$, where $term_i \in Label$ and $term_j \in Term$.
- Beta threshold (β) such that $\beta \leq P_C(term_i|term_j) \leq \alpha$, where $term_i \in Label$ and $term_j \in Term$.

These two thresholds are user-defined with a range of $[0, 1]$. Based on these thresholds, two sets of terms are identified (Fig.2):

- Alpha set ($\omega_\alpha^{(term_i)}$) is the set of terms for each label such that:

$$\omega_\alpha^{(term_i)} = \{term_j | \forall term_j \in Term : P_C(term_i|term_j) > \alpha\} \quad (2)$$

  i.e. the set of terms $term_j$ that co-occur with $term_i \in Label$ with a co-occurrence proportion higher than the threshold α.

- Beta set ($\omega_\beta^{(term_i)}$) is the set of terms for each label such that:

$$\omega_\beta^{(term_i)} = \{term_j | \forall term_j \in Term : \beta \leq P_C(term_i|term_j) \leq \alpha\} \quad (3)$$

  i.e. the set of terms that co-occur with $term_i \in Label$ with a co-occurrence proportion higher or equal than the threshold β and lower than the threshold α.

Table 1: Classification Model Concepts

DL concepts	Description
$Item \sqsubseteq \exists hasTerm.Term$	Items to classify,e.g. documents
$Term \sqsubseteq asString.String$	Terms (e.g. words) extracted from items
$Label \sqsubseteq Term$	Labels are terms used to classify items
$Label \sqsubseteq \forall broader.Label$	Broader relation between labels
$Label \sqsubseteq \forall narrower.Label$	Narrower relation between labels
$broader \equiv narrower^{-}$	Broader and Narrower are inverse relations
$Item \sqcap Term \equiv \perp$	Items and Terms are disjoint
$Item \equiv \exists hasTerm.Term$	Relation that links data items to the terms
$Label \sqsubseteq \forall hasAlpha.Term$	Terms used to create Alpha rules
$Label \sqsubseteq \forall hasBeta.Term$	Terms used to create Beta rules
$Item \sqsubseteq \exists isClassified.Label$	Relation that links items to labels

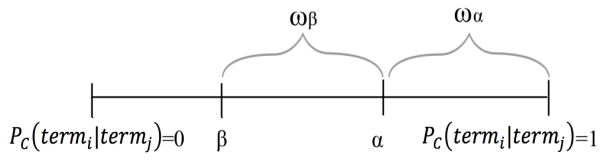


Figure 2: Alpha and Beta sets

Regarding the existence of Alpha and Beta sets for each item, four item categories are identified:

- Beta Empty set: $|\omega_\alpha^{(term_i)}| > 0 \wedge |\omega_\beta^{(term_i)}| = 0$
- Alpha Empty set: $|\omega_\alpha^{(term_i)}| = 0 \wedge |\omega_\beta^{(Label_i)}| > 0$
- Alpha and Beta not Empty set: $|\omega_\alpha^{(term_i)}| > 0 \wedge |\omega_\beta^{(term_i)}| > 0$
- Alpha and Beta Empty set: $|\omega_\alpha^{(term_i)}| = 0 \wedge |\omega_\beta^{(term_i)}| = 0$

Rules are created for the first three categories as follows. In an empty beta category only the ω_α is considered. Items are classified with labels if:

$$\forall label \forall item \exists term : hasTerm(item, term) \wedge term \in \omega_\alpha^{label} \rightarrow isClassified(item, label) \quad (4)$$

i.e. if the item has at least one term in $\omega_\alpha^{(term_i)}$ it is classified with $term_i$, $term_i \in Label$. For each term that complies with the above rule, a SWRL rule is created. For example, for a $|\omega_\alpha^{(term_i)}| = |\{t_1, t_2\}|$, the generated SWRL rules are presented in Table 2. In empty alpha category only the ω_β is considered. Items are classified with labels if:

$$\forall label \forall item : |\{\forall term : hasTerm(item, term) \wedge term \in \omega_\beta^{label}\}| \geq \delta \rightarrow isClassified(item, label) \quad (5)$$

i.e. if the item has at least δ terms in $\omega_\beta^{(term_i)}$, it is classified with $term_i$, $term_i \in Label$. One SWRL rule is generated for each combination of $term_j \in \omega_\beta^{(term_i)}$, where the number of combined terms is at least $\delta = \lceil |\omega_\beta^{(term_i)}| * p \rceil$, and $0 \leq p \leq 0.5$. For example, for a $|\omega_\beta^{(term_i)}| = |\{t_1, t_2, t_3\}| = 3$ and $p = 0.5$ resulting in $\delta = \lceil 3 * 0.5 \rceil = 2$, the generated SWRL rules are presented in Table 3. The set of generated beta rules is the combination C_n^m of m terms of a larger set of n elements. Regarding our approach, n is the number of possible terms $|\omega_\beta^{(term_i)}|$, and m the minimum number of terms δ in each rule (e.g. $C_{20}^{10} = 184756$). In order to limit the number of rules for each label we fix the value of $n \leq 10$. The terms are selected by ranking the terms in $\omega_\beta^{(term_i)}$ using the conditional proportion $P_C(term_i|term_j)$ as the ranking score.

Notice that the rules, which encompass more than δ terms, are not necessary because the combination of any δ terms is sufficient to classify the item. In non-empty alpha and beta category, beta and alpha rules are both considered. Alpha rules are evaluated as presented in the empty beta category. Beta rules are evaluated as presented in the empty alpha category but with a value $q = p * 2$ because beta rules are, by definition, less relevant than alpha rules. It corresponds to $\delta = \lceil |\omega_\beta^{(term_i)}| * q \rceil$, with $0 \leq q \leq 1$ and $q = p * 2$.

For the concepts in the fourth category (Alpha and Beta Empty) no enrichment rules are created because the cardinality of the sets is zero. The result of the resolution phase is the set of all the necessary and sufficient rules to classify an item in label.

3.2 Realization

The realization step includes two sub-steps: population and classification. The ontology-described knowledge base is populated with new items and their relevant terms

Table 2: Generated Alpha Rules (Example)

Alpha rules
$Item(?it), Term(?t_1), Label(?t_1), hasTerm(?it, ?t_1) \rightarrow isClassified(?it, ?t_1)$
$Item(?it), Term(?t_2), Label(?t_2), hasTerm(?it, ?t_2) \rightarrow isClassified(?it, ?t_2)$

Table 3: Generated Beta Rules (Example)

Beta rules
$Item(?it), Term(?t_1), Term(?t_2), Label(?t_3), hasTerm(?it, ?t_1), hasTerm(?it, ?t_2) \rightarrow isClassified(?it, ?t_3)$
$Item(?it), Term(?t_1), Term(?t_3), Label(?t_2), hasTerm(?it, ?t_1), hasTerm(?it, ?t_3) \rightarrow isClassified(?it, ?t_2)$
$Item(?it), Term(?t_2), Term(?t_3), Label(?t_1), hasTerm(?it, ?t_2), hasTerm(?it, ?t_3) \rightarrow isClassified(?it, ?t_1)$

at the assertion level (Abox). Each item is described with a set of relevant terms $\omega_{\gamma}^{(item_i)}$ such that:

$$\omega_{\gamma}^{(item_i)} = \{term_j | \forall term_j \in Term \wedge \gamma < tfidf_{(item_i, term_j, C)}\} \quad (6)$$

where γ is the relevance threshold, $\gamma < tfidf_{(item_i, term_j, C)}$, $term_j \in Term$, $item_i \in Item$ and $tfidf$ as calculated in the Vectorization step.

The classification sub-step performs the multi-label hierarchical classification of the items. Out-of-the-box tableaux-based or resolution-based reasoners such as Pellet [28], FaCT++ [32], or Hermit [27] are sound and complete to high expressive ontology, such as OWL2 SROIQ(D), but on the other hand they are not highly scalable and cannot handle huge volumes of data. Instead, we propose to use the rule-based reasoning that is less expressive but scales better. the rule-based reasoning exhaustively applies a set of rules to a set of triples (i.e. the data items) to infer conclusions [38], i.e. the item's classifications.

The rule-based inference engine uses rules to infer the subsumption hierarchy (i.e. concept expression subsumption) of the ontology and the most specific concepts for each data item. This leads to a multi-label classification of the items based in a hierarchical structure of the labels (Hierarchical Multi-label Classification). To infer the most specific labels, the rules generated in the resolution step are used. In addition, the following SWRL rule is used to classify an item with any subsuming label:

$$\begin{gathered} Item(?item), Label(?labelA), Label(?labelB), \\ broader(?labelA, ?labelB), \\ isClassified(?item, ?labelA) \\ \rightarrow isClassified(?item, ?labelB) \end{gathered} \quad (7)$$

These rules can be applied in a forward-chaining (i.e. materialization) or backward-chaining way. Based in these two types of rule-based reasoning, two types of classification are proposed: classification before query time and classification at query time.

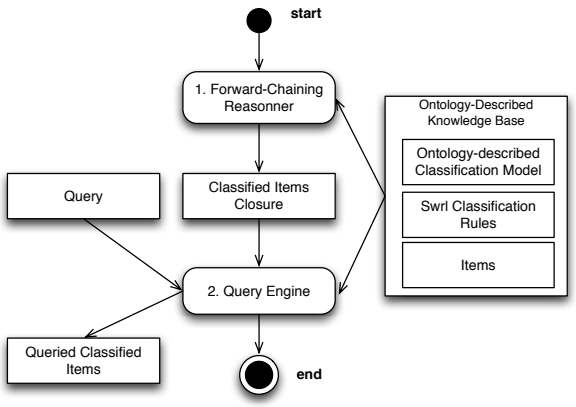


Figure 3: Classification before query time

Classification before query time (Fig. 3) is performed using a forward-chaining inference engine to create a closure with all inferred data, i.e. the inference rules are applied over the entire ontology-described knowledge base until all possible data is derived and materialized. After the closure has been calculated, the query engine directly queries the closure retrieving the classifications very fast. On the other hand, the closure must be updated for every change in the ontology-described knowledge base. Therefore, creating a closure of inferred data can be expensive due to the data volume, velocity of changes, and the quantity and complexity of rules.

Classification on query time (Fig. 4) is performed by backward-chaining inference applying the rules only over the strictly necessary data to answer the query. By applying the rules over the strictly necessary data has the advantage of addressing the rapidly changing data feature of Big Data. On the other hand, the main disadvantage is the need to activate the inference engine for each query, which is affected by the volume and quantity and expressiveness of rules.

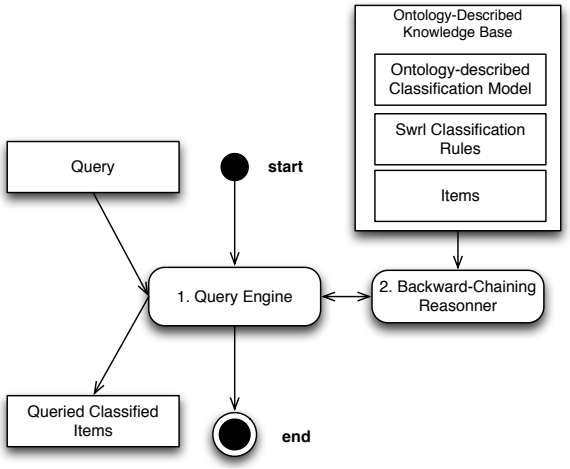


Figure 4: Classification on query time

Despite both types of classification can be used in the Semantic HMC process, a carefully combination of both processes is necessary due to the type of use cases of the system (i.e. retrieve all data or parts of data).

4 IMPLEMENTATION

This section describes the implementation of the proposed hierarchical multi-label classification process. The process is implemented as a combination of available Java libraries that natively support parts of the process.

In the first three steps (indexation, vectorization and hierarchization) of the Semantic HMC process, Big Data technologies are used, including MapReduce [6]. MapReduce is a programming model, which addresses large scale data processing on several machines. In the MapReduce paradigm [6], users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. The MapReduce algorithms are deployed on a Hadoop cluster (https://hadoop.apache.org/). We choose Hadoop because of its open-source nature and its ability for integration with the previously used tools. The vectors, the co-occurrence matrix and the hierarchy are stored in HDFS (Hadoop distributed file system), which is used during the resolution and realization steps.

The next subsections describe the implementation details of each step of the classification process.

4.1 Resolution

The resolution process creates the ontology rules used to relate the labels and the data items. We assume that α and β thresholds are user-defined settings. The rule creation process is divided in a sub-process for each $label_i \in Label$. In each sub-process, $\omega_\alpha^{(label_i)}$ and $\omega_\beta^{(label_i)}$ sets are calculated using the co-occurrence matrix, then classification rules are created for each label. Exploiting a huge co-occurrence matrix to create the ontology rules is a very intensive task, thus this process is also distributed to several machines in the MapReduce paradigm. A MapReduce job creates the rules from the co-occurrence matrix.

The set of pairs $< (term_i, term_j), P(x|y) >$ is used as the input of the map function. The $(key, value)$ pairs are defined as:

- key is a tuple $(term_i, term_j)$ where both $term_i$ and $term_j$ are terms identified in the Vectorization step.
- $value$ is the proportion $P(x|y)$

In the map phase, the α and β thresholds are applied to the proportion $P(x|y)$ of each pair $< (term_i, term_j), P(x|y) >$ where $term_i \in \omega_{lT}$ or $term_j \in \omega_{lT}$. The map function outputs a list of $< (RuleType, label_i), term_j >$ pairs where:

- RuleType is a descriptor for the type of rule (alpha or beta)
- $label_i$ is the label related by the new rule
- $term_j$ is a term used to relate items with $label_i$ that can be the term of an alpha rule, or a term comprised in a beta rule

According to the MapReduce paradigm, pairs are shuffled by key (i.e.$(RuleType,\ label_i)$) and the MapReduce reduce function is executed for each set of pairs with the same key. The reduce function aggregates the rules by $label_i$ and outputs the set of alpha terms $\omega_\alpha^{(term_i)}$ and beta terms $\omega_\beta^{(term_i)}$ for each $label_i$. The rules are serialized in SWRL language and stored in the ontology-described knowledge base using the OWL-API library.

The generated rules along with the label hierarchy are used in the realization step to classify new items.

4.2 Realization

The realization step populates the ontology and performs the multi-label hierarchical classification of the items.

First the ontology is populated with new items and the most relevant terms to describe each document in

an assertion level (Abox). The $tfidf$ vectors for each document calculated in vectorization allow measuring the relevancy of a term in a text document (item) and calculate the set of relevant terms $\omega_{\gamma}^{(item_i)}$. To store, manage and query the ontology-described knowledge base (Tbox+Abox) a triple-store is used. Because highly expressive forward chaining description logics reasonners do not scale well and, on the other hand, because Web-Scale Reasoners based on MapReduce programming model, like WebPie, are limited to low expressive ontologies as OWL-Horst fragment, in our preliminary prototype we decided to adopt the classification at query time by using a triple-store with a backward-chaining inference engine.

Due to backward-chaining query performance issues identified in [9] a rule selection approach was developed to execute only the rules needed to classify the items for that query. Two main query types are identified: (1) retrieve all items classified with a label and (2) retrieve all labels that classifies an item.

To retrieve all items classified with a label $label_i$ only the rules with $label_i$ in the rule head (i.e. $isClassified(?item, label_i)$) are activated. To retrieve the labels that classifies an item $item_i$ only the rules with at least one term $term_i \in \omega_{\gamma}^{(item_i)}$ in the rule tail (i.e. $(hasTerm(?item, term_i))$ are activated.

The OWL-API library is used to populate the OWL ontology with new items. A scalable triple-store called Stardog (http://docs.stardog.com) is used to store and query the ontology-described Knowledge Base (Tbox+Abox). Stardog is also used to perform reasoning by backward-chaining inference as well as SWRL rules inference. The rule selector was developed in java, and interacts with Stardog to optimize the query performance.

5 EXPERIMENTS

In this section we evaluate the classification performance of the Semantic HMC process for unstructured text classification in a Big Data context. First preliminary results of the proposed classification process are discussed regarding a large dataset. Second the quality of the results is evaluated. Finally we discuss the obtained results regarding some algorithms from the state of the art.

5.1 Experiments with Large Datasets

In this subsection the preliminary results of the proposed classification process are discussed regarding a large dataset of unstructured text documents. First the dataset, the environment and the settings used to test the process are described. Then the experimental results are presented and discussed.

Table 4: Wikipedia-based DataSets

Dataset	Number of articles	Size (GB)
Wikipedia 1	174900	1.65
Wikipedia 2	407000	2.21
Wikipedia 3	994000	5

Table 5: Execution Settings

Parameter	Step	Value
Alpha Threshold	Resolution	90
Beta Threshold	Resolution	80
Term ranking (n)	Resolution	5
p	Resolution	0.25
Term Threshold (γ)	Realization	2

5.1.1 Test Environement

The dataset is composed of unstructured text articles. The articles are extracted from dumps of the French version of Wikipedia with different sizes as described in Table 4.

Some thresholds and settings used in the process have a strong impact on the results. Table 5 shows the different parameters and their values used in preliminary results. The same values are used for all datasets. The co-occurrence matrix and the hierarchy calculated on [25] are used as input where the number of terms, labels, and subsumption relations are presented in Table 6.

5.1.2 Results

The aim of the preliminary test is to check the scalability of the system according to the number of items from the same dataset. For that we monitor: (1) The number of learned classification rules (i.e. α and β rules); (2) The number of classifications (i.e. isClassified relations) from each sub-dataset.

In the previous work [25] it was demonstrated that the number of labels decreases when the size of the dataset grows. The number of learned classification rules (α and β rules) for each sub-dataset is depicted in Fig. 5. The reader can observe a decrease in the number of learned

Table 6: Previous Results

Dataset	Wiki 1	Wiki 2	Wiki 3
Number of Terms	10973	13053	23859
Number of Labels	3680	1981	1545
Number of relations	10765	2754	1315

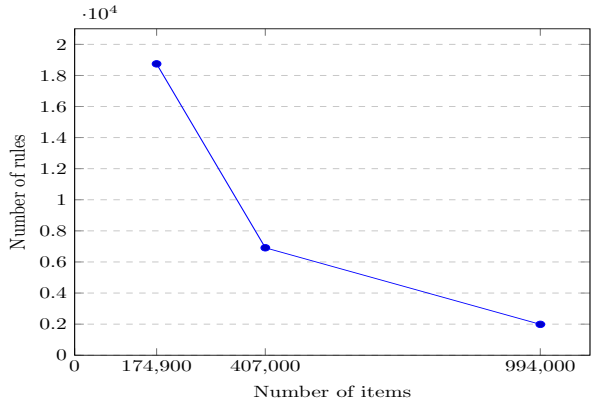


Figure 5: Number of learned rules according to each dataset

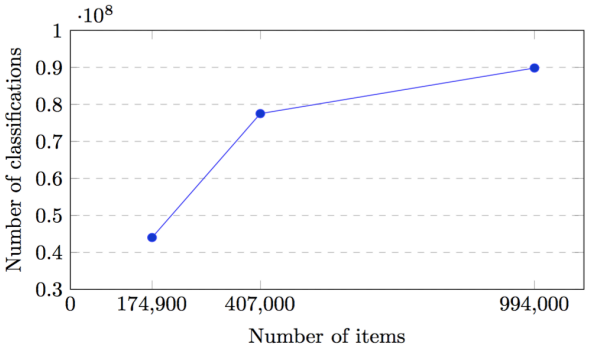


Figure 6: Number of classifications (learned isClassified relations) according to each dataset

rules as a consequence of the decrease of the number of learned labels. The number of classification relations (isClassified) for each sub-dataset is depicted in Fig. 6. The reader can observe an increase in the number of classifications while the size of the dataset grows even if the number of rules decrease.

5.2 Quality Evaluation

In this subsection we evaluate the classification performance of the Semantic HMC process for unstructured text classification in a Big Data context. First the dataset, the test environment and the experimental settings used to evaluate the process are described. Then the experimental results are presented and discussed.

The evaluation is done using a pre-labeled dataset, composed of training and test data. The training set is used to learn hierarchical relations between the pre-defined labels and classification rules. The test set is used to calculate the classification performance of the algorithm based on standard quality metrics (i.e. precision, recall, F-measure).

Table 7: Delicious dataset specifications

$\|Train\|$	$\|Test\|$	$\|Labels\|$	$\|Terms\|$
12,910	3,181	983	500

To be able to compare our approach with state-of-the-art, we use a pre-defined set of labels instead of automatically learned labels as it is described in the previous experiment

5.2.1 Delicious dataset

The Delicious dataset[2] is used to perform this evaluation. This dataset is composed of labeled textual data from web pages extracted from the Delicious social bookmarking website [34]. Table 7 shows the dataset specifications. The Delicious dataset was chosen because it contains very few features (words) compared to the number of labels, rendering accurate classification difficult [23]. Also, it has been used to evaluate several multi-label classification systems, thus it provides a good baseline to compare our approach.

5.2.2 Metrics

To evaluate the quality of the SHMC process we use previous studies in the HMC evaluation as reference [3, 39, 4]. Precision and recall are two standard metrics widely used in text categorization literature to evaluate the generalization performance of the learning system on a given category. For single-label classification problems, precision and recall are defined as:

$$Precision = \frac{TP}{(TP + FP)} \tag{8}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{9}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. However, performance evaluation on multi-label learning algorithms is not trivial as each item is associated with multiple labels simultaneously, where traditional single-label criteria such as accuracy, precision or recall cannot be directly applied. Specific evaluation metrics to multi-label learning are proposed in literature and generally categorized into two groups [44]:

- Example-based metrics evaluating the learning system's performance on each test example

[2] http://mulan.sourceforge.net/datasets-mlc.html

(labelled item) separately, and then returning the mean value across the test set.

- Label-based metrics learning system's performance on each class label separately, and then returning the macro/micro-averaged value across all class labels.

We use a label-based metric to evaluate the Semantic HMC. In label-based metrics the micro-averaged precision and recall are calculated as [17]:

$$Precision_{micro} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i} \quad (10)$$

$$Recall_{micro} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i} \quad (11)$$

and the macro-averaged precision and recall are calculated as:

$$Precision_{macro} = \frac{1}{n}\sum_{i}^{n}(\frac{TP_i}{(TP_i + FP_i}) \quad (12)$$

$$Recall_{macro} = \frac{1}{n}\sum_{i}^{n}(\frac{TP_i}{TP_i + FN_i}) \quad (13)$$

where TP_i is the number of true positives, FP_i the number of false positives and FN_i the number of false negatives for $Label_i$. Another standard evaluation metric for classification is the $F1$ measure that is the harmonic mean of precision and recall. The $F1$ is calculated for micro-average or macro-average, defined as follows:

$$MacroF1 = \frac{2 * (Precision_{macro} * Recall_{macro})}{(Precision_{macro} + Recall_{macro})} \quad (14)$$

$$MircroF1 = \frac{2 * (Precision_{micro} * Recall_{micro})}{(Precision_{micro} + Recall_{micro})} \quad (15)$$

5.2.3 Results

The Hierarchization phase of the Semantic HMC process automatically generates a hierarchical relations between labels. This hierarchy along with the classification rules created in the resolution step are used to perform hierarchical multi-label classification. Figure 7 shows a sample of the hierarchichal relations ($skos$: $hasBroaderRelation$) between labels automatically created for the Delicious dataset. The set of parameters used to create the hierarchy and classification rules is described in table 8. This parameters can have a high impact in the quality of the results. The top and bottom tresholds are used to calculate the hierarchichal relations between labels as defined in [25].

Table 8: Execution Settings for Delicious Dataset

Parameter	Step	Value
Top Threshold	Hierarchization	50
Bottom Threshold	Hierarchization	40
Alpha Threshold	Resolution	20
Beta Threshold	Resolution	10
Term ranking (n)	Resolution	5
p	Resolution	0.25
Term Threshold (γ)	Realization	2

Table 9: Quality results for the Delicious Dataset

	Precision	Recall	F1-measure
Micro	0.284	0.74	0.410
Macro	0.0676	0.178	0.0979

Table 9 shows the results obtained by the Semantic HMC process on the Delicious dataset. The micro averaged precision and recall are higher than the macro averaged precision and recall. Also the precision is lower than the recall in both cases.

5.3 Comparison with the State of the Art

Table 10 shows the Macro-F1 measure and Micro-F1 measure obtained on the Delicious dataset. The results of the proposed process (SHMC) are compared with several state-of-the-art approaches results with the same dataset [19, 29, 23]. The state-of-the-art approaches used for comparison are: Binary relevance (BR) [33], Classifier chaining (CC) [26], TNBCC [29], Path-BCC [29], Hierarchy Of Multi-label Classifiers (HOMER) [34] , Multi-label k-nearest neighbors (ML-kNN) [43], Random forest of predictive clustering trees (RF-PCT) [18], Random forest of ML-C4.5 (RFML-C4.5) [18] and the Collapsed Gibbs Sampling (CGS) algorithm [12].

In Table 10 it is observed that the Semantic HMC approach outperforms state-of-the-art approaches in micro F1-measure, while the macro F1-measure is comparable to most other approaches. These results show that the classification performance of our ontology-based approach is comparable to the performance of the selected algorithms from the state-of-the-art in machine learning.

6 Conclusions

This paper describes in detail an unsupervised hierarchical multi-label classification process for unstructured text in the scope of Big Data. The label

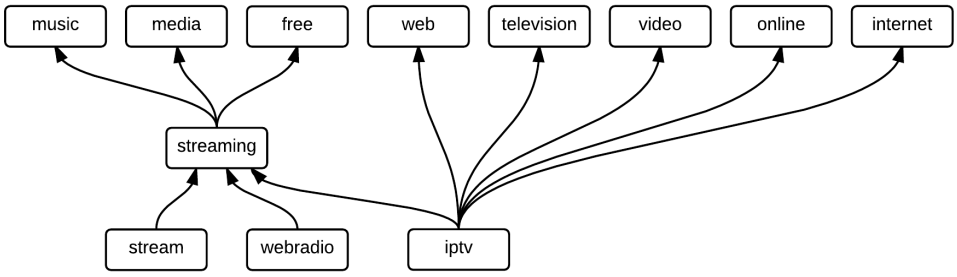


Figure 7: Automatically generated hierarchy from Delicious dataset (sample)

Table 10: Performance of various algorithms on the Delicious dataset

Algorithm	Macro F1	Micro F1
BR [33]	0.096	0.234
CC [26]	0.100	0.236
CGS_p [12]	0.103	0.297
HOMER [34]	0.103	0.339
ML-kNN [43]	0.051	0.175
Path-BCC [29]	0.084	N/A
RF-PCT [18]	0.083	0.248
RFML-C4.5 [18]	**0.142**	0.269
SHMC	0.097	**0.410**
TNBCC [29]	0.0880	N/A

hierarchy is first automatically obtained and used as the first input for the ontology construction. Then for each label a set of rules is created to relate the data items with the taxonomy concepts. Finally, the learned ontology is populated with the data items resulting in an ontology-described classification model. To classify the items with labels a rule-based web reasoner is used. Due to the limitations of reasoners, only the classification on query time was considered, experimented and evaluated. The process prototype was successfully implemented in a scalable and distributed platform to process Big Data.

The experimental evaluation highlights three aspects of the Semantic HMC process. First, the process can learn classification rules from huge amount of data and classify documents automatically. Secondly, evaluation results prove that the classification performance of the Semantic HMC process that uses ontologies and rule-based reasoning to classify unstructured text documents is comparable to the performance of algorithms from the state-of-the-art in the field of machine learning.

Finally, unlike most approaches from the data-mining field, the ontology-based approach provides human-readable explanations of the classifications, which can be used to monitor the classification process by experts.

Our current work is twofold: (1) the application of the process to domain-specific data and (2) the maintenance of the classification model regarding a stream of data in a Big Data Context.

ACKNOWLEDGEMENTS

This project is funded by the company Actualis SARL, the Bourgogne Franche-comté region, the French agency ANRT and through the Portuguese COMPETE Program under the project AAL4ALL (QREN13852). We would like to thank GRAPHIQ (https://www.graphiq.com/), which sponsors the open-access publishing of this paper.

REFERENCES

[1] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, "On Smoothing and Inference for Topic Models," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 27–34.

[2] D. Ben-David, T. Domany, and A. Tarem, "Enterprise Data Classification Using Semantic Web Technologies," in *The Semantic Web–ISWC 2010*, ser. ISWC'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 66–81.

[3] W. Bi and J. Kwok, "Multi-Label Classification on Tree- and DAG-Structured Hierarchies," *Yeast*, pp. 1–8, 2011.

[4] R. Cerri, R. C. Barros, and A. C. de Carvalho, "Hierarchical Multi-label Classification Using Local Neural Networks," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 39–56, 2014.

[5] M. Chen, S. Mao, and Y. Liu, "Big Data: A Survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, January, 2014.

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 1–13, 2008.

[7] Z. Elberrichi, B. Amel, and T. Malika, "Medical Documents Classification Based on the Domain Ontology MeSH," *arXiv preprint arXiv:1207.0446*, 2012.

[8] J. Fang, L. Guo, and Y. Niu, "Documents Classification by Using Ontology Reasoning and Similarity Measure," in *Proceedings - 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010*, vol. 4, 2010, pp. 1535–1539.

[9] T. M. Farias, A. Roxin, and C. Nicolle, "FOWLA, A Federated Architecture for Ontologies," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9202, 2015, pp. 97–111.

[10] D. Fensel, *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Heidelberg, Germany: Springer-Verlag, 2001.

[11] A. Galinina and A. Borisov, "Knowledge Modelling for Ontology-Based Multiattribute Classification System," *Applied Information and Communication*, pp. 103–109, 2013.

[12] T. L. Griffiths and M. Steyvers, "Finding Scientific Topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[13] T. Hassan, R. Peixoto, C. Cruz, A. Bertaux, and N. Silva, "Semantic HMC for Big Data Analysis," in *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*, 2015, pp. 26–28.

[14] P. Hitzler and K. Janowicz, "Linked Data, Big Data, and the 4th Paradigm," *Semantic Web*, vol. 4, no. 3, pp. 333–335, 2013.

[15] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," *Syntax*, vol. 21, no. May, p. 79, 2004.

[16] I. Johnson, J. Abécassis, B. Charnomordic, S. Destercke, and R. Thomopoulos, "Making Ontology-Based Knowledge and Decision Trees Interact: An Approach to Enrich Knowledge and Increase Expert Confidence in Data-Driven Models," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, 2010, vol. 6291 LNAI, pp. 304–316.

[17] D. Kocev, S. Džeroski, G. Madjarov, and D. Gjorgjevikj, "An Extensive Experimental Comparison of Methods for Multi-Label Learning," in *Pattern Recognition*, vol. 45, no. 9, 2012, pp. 3084–3104.

[18] D. Kocev, C. Vens, J. Struyf, and S. Džeroski, "Ensembles of Multi-Objective Decision Trees," in *European Conference on Machine Learning*. Springer, 2007, pp. 624–631.

[19] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition*, vol. 45, no. 9, pp. 3084–3104, 2012.

[20] R. Moller and V. Haarslev, "Tableau-Based Reasoning," in *Handbook on Ontologies*, 2009, p. 654.

[21] B. Motik and U. Sattler, "A Comparison of Reasoning Techniques for Querying Large Description Logic Aboxes," in *Logic for Programming, Artificial Intelligence, and Reasoning*. Springer, 2006, pp. 227–241.

[22] L. Obrst, "Ontologies for Semantically Interoperable Systems," in *Proceedings of the twelfth international conference on Information and knowledge management - CIKM '03*. ACM Press, 2003, pp. 366–369.

[23] Y. Papanikolaou, T. N. Rubin, and G. Tsoumakas, "Improving Gibbs Sampling Predictions on Unseen Data for Latent Dirichlet Allocation," *arXiv preprint arXiv:1505.02065*, 2015.

[24] R. Peixoto, T. Hassan, C. Cruz, A. Bertaux, and N. Silva, "An Unsupervised Classification Process for Large Datasets Using Web Reasoning," in *SBD'16: Semantic Big Data Proceedings*, ACM, Ed., San Francisco (CA), USA, 2016.

[25] R. Peixoto, T. Hassan, C. Cruz, A. Bertaux, and N. Silva, "Semantic HMC: A Predictive Model using Multi-Label Classification For Big Data," in *The 9th IEEE International Conference on Big Data Science and Engineering (IEEE BigDataSE-15)*, 2015.

[26] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier Chains for Multi-Label Classification," *Machine learning*, vol. 85, no. 3, pp. 333–359, 2011.

[27] R. Shearer and I. Horrocks, "Hypertableau Reasoning for Description Logics," *Journal of Artificial Intelligence Research*, vol. 36, pp. 165–228, 2009.

[28] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A Practical OWL-DL Reasoner," in *Web Semantics*, vol. 5, no. 2, 2007, pp. 51–53.

[29] L. E. Sucar, C. Bielza, E. F. Morales, P. Hernandez-Leal, J. H. Zaragoza, and P. Larrañaga, "Multi-label Classification with Bayesian Network-based Chain Classifiers," *Pattern Recognition Letters*, vol. 41, pp. 14 – 22, 2014.

[30] A. R. Syed, K. Gillela, and C. Venugopal, "The Future Revolution on Big Data," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 6, pp. 2446–2451, 2013.

[31] H. J. Ter Horst, "Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary," *Web Semantics*, vol. 3, no. 2-3, pp. 79–115, 2005.

[32] D. Tsarkov and I. Horrocks, "FaCT++ Description Logic Reasoner: System Description," in *Proceedings of the Third International Joint Conference (IJCAR)*, U. Furbach and N. Shankar, Eds., vol. 4130. Springer Berlin / Heidelberg, 2006, pp. 292–297.

[33] G. Tsoumakas, I. Katakis, and A. Overview, "Multi-Label Classification : An Overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. September, pp. 1–13, 2007.

[34] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and Efficient Multilabel Classification in Domains with Large Number of Labels," in *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, 2008, pp. 30–44.

[35] J. Urbani, "Three Laws Learned from Web-scale Reasoning," in *2013 AAAI Fall Symposium Series*, 2013, pp. 76–79.

[36] J. Urbani, S. Kotoulas, J. Maassen, F. Van Harmelen, and H. Bal, "OWL reasoning with WebPIE: Calculating the closure of 100 billion triples," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Berlin Heidelberg, 2010, vol. 6088 LNCS, no. PART 1, pp. 213–227.

[37] J. Urbani, S. Kotoulas, J. Maassen, F. Van Harmelen, and H. Bal, "WebPIE: A Web-scale Parallel Inference Engine using MapReduce," *Journal of Web Semantics*, vol. 10, pp. 59–75, 2012.

[38] J. Urbani, F. Van Harmelen, S. Schlobach, and H. Bal, "QueryPIE: Backward Reasoning for OWL Horst Over Very Large Knowledge Bases," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, ser. ISWC'11, vol. 7031 LNCS, no. PART 1. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 730–745.

[39] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision Trees for Hierarchical Multi-Label Classification," in *Machine Learning*, vol. 73, no. 2, 2008, pp. 185–214.

[40] S. Vogrincic and Z. Bosnic, "Ontology-Based Multi-Label Classification of Economic Articles," *Comput. Sci. Inf. Syst.*, vol. 8, no. 1, pp. 101–119, 2011.

[41] D. Werner, N. Silva, C. Cruz, and A. Bertaux, "Using DL-Reasoner for Hierarchical Multilabel Classification Applied to Economical E-News," in *Proceedings of 2014 Science and Information Conference, SAI 2014*, 2014, pp. 313–320.

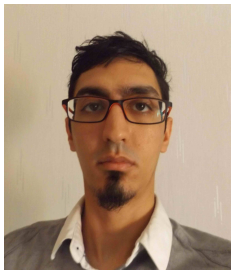
[42] H. Wu, J. Liu, D. Ye, H. Zhong, and J. Wei, "A Distributed Rule Execution Mechanism Based on MapReduce in Sematic Web Reasoning," *Proceedings of the 5th Asia-Pacific Symposium on Internetware - Internetware '13*, pp. 1–7, 2013.

[43] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A Lazy Learning Approach to Multi-Label Learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[44] M. L. Zhang and Z. H. Zhou, "A Review on Multi-Label Learning Algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

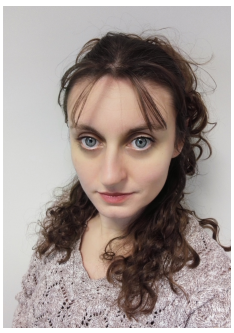
AUTHOR BIOGRAPHIES

Rafael Peixoto is affiliated with Instituto Superior de Engenharia do Porto (ISEP) - Polytechnic of Porto as a Fellowship Researcher since 2010 and a PhD student at the Univsity of Burgundy in France and affiliated to the laboratory LE2I (Electronics, Image and Computer Science) since 2013. He received its bachelor degree in Computer Science Engineering from the School of Engineering of the Polytechnic of Porto in 2010 and also the MSc degree in Computer Science from the same institution in 2012. His main interests are Data Science, Big Data and Semantic Web.



Thomas Hassan is a PhD student at the laboratory LE2I (Electronics, Image and Computer Science) since October 2014 and Teacher at the University of Burgundy (France) since January 2015. He obtained a Master in Computer science from the University of Burgundy in July 2014. His main research areas are Big Data and Semantic Web. He works more specifically on the ontology learning in a Big Data context, and applications to news recommendation.

Dr. Christophe Cruz is an Associate Professor at the laboratory LE2I (Electronics, Image and Computer Science) since September 2005 - Université Bourgogne-Franche-Comté (France) where he obtained a PhD in Computer science in 2004. He is responsible for the cluster team #3: Intelligent environments of the laboratory LE2I. His main research areas are semantic modeling, knowledge representation and reasoning.



Dr. Aurélie Bertaux earned her PhD in Computer Science in 2010. In 2013 she became associate professor in LE2I UMR CNRS 6306 (Laboratory of Electronics, Informatics and Image) of the University of Burgundy in France. Her research stands in Data Mining, especially in Formal Concept Analysis (developed during her phD) and in Graph Mining (developed during a post doc in Grenoble Informatics Laboratory, France). Now, working in CheckSem research team, she starts to develop skills in the field of the team: ontologies, and to build bridges between those three domains.

Dr. Nuno Silva is a professor and researcher in the Knowledge Engineering and Decision Support Research Center (GECAD) of the School of Engineering at the Polytechnic of Porto. His research areas include Information Integration, Knowledge Engineering and the Semantic Web. He managed several R&D projects in these fields and supervised 4 PhDs. Silva has a PhD in computer science from the University of Trás-os-Montes and Alto Douro, Portugal. Contact him at nps@isep.ipp.pt.