An unsupervised classification process for large datasets using web reasoning

Rafael Peixoto[1,2], Thomas Hassan[1], Christophe Cruz[1], Aurélie Bertaux[1], Nuno Silva[2]

[1] LE2I UMR6306, CNRS, Arts et Métiers, Univ. Bourgogne Franche-Comté, F-21000 Dijon, France
{thomas.hassan, christophe.cruz, aurelie.bertaux}@u-bourgogne.fr
[2] GECAD - ISEP, Polytechnic of Porto, Porto, Portugal, {rafpp,nps}@isep.ipp.pt

ABSTRACT

Determining valuable data among large volumes of data is one of the main challenges in Big Data. We aim to extract knowledge from these sources using a Hierarchical Multi-Label Classification process called Semantic HMC. This process automatically learns a label hierarchy and classifies items from very large data sources. Five steps compose the Semantic HMC process: Indexation, Vectorization, Hierarchization, Resolution and Realization. The first three steps construct automatically the label hierarchy from data sources. The last two steps classify new items according to the label hierarchy. This paper focuses in the last two steps and presents a new highly scalable process to classify items from huge sets of unstructured text by using ontologies and rule-based reasoning. The process is implemented in a scalable and distributed platform to process Big Data and some results are discussed.

CCS Concepts

• Information systems→Ontologies • Information systems→Clustering and classification • Computer systems organization→Distributed architectures • Computing methodologies→Knowledge representation and reasoning

Keywords

classification; Big-Data; ontology; machine learning

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 [3, 11]: 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.[3]. 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 [20] such as audio, video and text. Value measures 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 [3].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
SBD'16, July 01 2016, San Francisco, CA, USA
© 2016 ACM. ISBN 978-1-4503-4299-5/16/07...$15.00
DOI: http://dx.doi.org/10.1145/2928294.2928301

Werner et al. [27] 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 uses an out-of-the-box Description Logics (DL) Web Reasoner like Pellet [14], FaCT++ [21], or Hermit [16]. Most of these reasoners are sound and complete to high expressiveness, as OWL2 SROIQ (D) expressiveness, but on the other hand they do not scale [27]. 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 [27] and to exploit value by analyzing Big Data using a Semantic Hierarchical Multi-Label Classification process (Semantic HMC) [10]. Hierarchical Multi-Label Classification (HMC) is the combination of Multi-Label classification and Hierarchical classification [2].

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 as 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 taxonomy is automatically obtained and used as the first input for the ontology construction [8]. Then, for each taxonomical concept (classification labels) a set of rules is created to relate the data items to the taxonomy concepts. Then the learned ontology is populated with the data items. Hence, Semantic HMC proposes five individually scalable steps to reach the aims of Big Data analytics [10].

- *Indexation* extracts terms from data items and creates an index of data items.
- *Vectorization* calculates the term-frequency vectors of the indexed items.
- *Hierarchization* creates the label taxonomy (i.e. subsumption hierarchy) using term-frequency vectors.
- *Resolution* creates the reasoning rules to relate data items with the labels based on term-frequency vectors.

- *Realization* first populates the ontology with items and then for each item determines the most specific label and all its subsuming labels.

The first three steps learn the label hierarchy from unstructured data as described in [19]. As a follow up, this paper focuses on the two last steps of the Semantic HMC process. It proposes a new process to hierarchically multi-classify items from huge sets of unstructured texts using DL ontologies and Rule-based reasoning. The process is implemented using scalable approaches that distribute the process by several machines in order to reach high performance and scalability required by Big Data.

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, is introduced some background and discuss the current related work about automatic hierarchical multi-label classification from unstructured text using DL ontologies and reasoning. The following subsections discuss the related work of ontologies and Web Reasoning in classification context.

2.1 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 that operate close to the human concept level bridging the gap between the human requirements and the computational requirements [18].

Galinina et al. [9] 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 domain description, ontologies can be used to improve the classification process. Elberrichi et al. [5] present a two-steps 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. [13] 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. [26] are concerned with automatically creating an ontology from text documents without any prior knowledge about their content.

2.2 Web reasoning in Classification context

Reasoning is used at ontology development or maintenance time as well as at the time ontologies are used for solving application problems [15]. Web reasoning can be used to improve the classification process. In [6] authors presents a document classification method that uses ontology reasoning and similarity measures to classify the documents. In [1] authors introduce a generic, automatic classification method that uses Semantic Web technologies to: 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 [27], the authors proposed 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 to reduce 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 [14], FaCT++ [21], or Hermit [16] is used.

2.3 Discussion

Most 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 [27] authors uses out-of-the-box reasoning to classify economical documents but their scalability is limited and cannot be used in large datasets as required in Big Data context. However as Semantic Web is growing, new high-performance Web Scale Reasoning methods have been proposed [24]. Rule-based reasoning approach allows the parallelization and distribution of work by large clusters of inexpensive machines by programming models for processing and generating large data sets as Map-reduce [4]. Web Scale Reasoners [24] however, instead of using traditional DL approaches like Tableau [14] [21], Resolution [17] or Hypertableau [16], use entailment rules for reasoning over ontologies. Web-Scale Reasoners based in Map-reduce programming model like WebPie [22] outperforms all other published approaches in an inference test over 100 billion triples [25]. Instead, recent implementations of Web-Scale Reasoners as WebPie are limited to low expressive ontologies as OWL-Horst fragment [12] due to the complexity of implementation and performance at web scale. In [28] 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 two last steps (Resolution and Realization) of the hierarchical multi-label classification process are described in detail.

In [19], 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.

Table1. Classification Model

DL concepts	Description
$Item \sqsubseteq \exists hasTerm.Term$	Items to classify (e.g. doc) has terms
$Term \sqsubseteq \top$	Terms (e.g. word) 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
$Item \sqcap Term = \emptyset$	Items and Terms are disjoint
$Item \sqsubseteq \forall isClassified.Label$	Relation that links items with labels

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 $item_1$ to be classified as $label_1$. The rules will define the necessary and sufficient terms of an item so the item is classified in label.

The rules creation process uses thresholds as proposed in [27] to select the necessary and sufficient terms. The main difference to such method is that instead of translating the rules into logical constraints of an ontology captured in Description Logic, these rules are translated into rules in the Semantic Web Rule Language (SWRL).

The main interest in using SWRL rules is to reduce the reasoning effort, thus improving the scalability and performance of the system. The aim is to use more but simpler SWRL rules that will be applied to the ontology in order to classify items.

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

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

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

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

  i.e. is the set of terms t_j that co-occur with $t_i \in Label$ with a co-occurrence proportion higher than the threshold α.
- Beta set ($\omega_\beta^{t_i}$) is the group of terms, for each label such that:

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

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

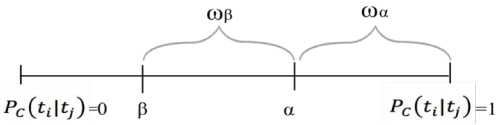

Figure 1. Alpha and beta Sets.

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

- Beta Empty such as: $|\omega_\alpha^{t_i}| > 0 \wedge |\omega_\beta^{t_i}| = 0$
- Alpha Empty such as: $|\omega_\alpha^{t_i}| = 0 \wedge |\omega_\beta^{t_i}| > 0$
- Alpha Beta not Empty such as: $|\omega_\alpha^{t_i}| > 0 \wedge |\omega_\beta^{t_i}| > 0$
- Alpha and Beta Empty such as: $|\omega_\alpha^{t_i}| = 0 \wedge |\omega_\beta^{t_i}| = 0$

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

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

I.e. if the item has at least one term in $\omega_\alpha^{t_i}$ it is classified with t_i, $t_i \in Label$. For each term that complies with the above rule, a SWRL rule is created. For example, for a $|\omega_\alpha^{t_i}| = |\{t_1, t_2\}|$, 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:

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

I.e. if the item has at least δ terms in $\omega_\alpha^{t_i}$ it is classified with $term_i$, $term_i \in$ Label. One SWRL rule is generated for each combination of $t_j \in \omega_\beta^{t_i}$ where the number of combined terms is at least $\delta = \lceil |\omega_\beta^{t_i}| * p \rceil$, and $0 \leq p \leq 0.5$. For example, for a $|\omega_\beta^{t_i}| = |\{t_1, t_2, t_3\}| = 3$ 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 $|\omega_\beta^{t_i}|$, and m the minimum number terms δ in each rule (e.g. $C_{20}^{10} = 184\,756$). In order to limit the number of rules for each label we fix the value of $n <= 10$. The terms are selected by ranking of the terms in $\omega_\beta^{t_i}$ using the conditional proportion $P_C(t_i|t_j)$ as score.

Table 2. Generated Alpha Rules (Example)

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

Table 3. Generated Beta Rules (Example)

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

Notice that the rules that 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 less relevant than alpha rules. It corresponds to $\delta = \left\lfloor |\omega_{\beta}^{t_i}| * q \right\rfloor$, 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 is populated with new items and their relevant terms in an assertion level (Abox). Each item is populated with a set of relevant terms $\omega_{\gamma}^{\text{item}_i}$ 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 relevancy threshold $\gamma < tfidf_{item_i,term_j,C}$, $term_j \in Term$, $item_i \in Item$ and $tfidf$ is the term frequency calculated in Vectorization step as described in [19].

The classification sub-step performs the multi-label hierarchical classification of the items. Out-of-the-box tableaux-based or resolution-based reasoner's such as Pellet [14], FaCT++ [21] or HermiT [16] are sound and complete to high expressive ontology but not highly scalable. Instead, we propose to use rule-based reasoning approach that is less expressive but scales better. Rule-based reasoning applies exhaustively a set of rules to a set of triples (i.e. the data items) to infer conclusions [23], 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 (i.e. realization of and individual). 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. To classify the item with the all its broader labels the following SWRL rule is used:

$$Item(?\,item), Label(?\,label), Label(?\,labelB), broader(?\,label, ?\,labelB), \ isClassified(?\,item, ?\,label) \rightarrow isClassified(?\,item, ?\,labelB)$$

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

Classification before query time is preformed 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. Once the closure is calculated, the query-time process is very simple and fast, but the closure must be updated at every change in the ontology described knowledge base. Therefore, creating a closure of inferred data can be expensive due to data volume, velocity of changes and quantity and complexity of rules.

Classification on query time 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 that for each query is always necessary to activate the inference engine, which is affected by the volume and quantity and expressivity of rules.

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 [4]. MapReduce is a programming model, which addresses large scale data processing on several machines. In the MapReduce paradigm, "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 will be 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 $\text{Label}_i \in$ Label. In each sub-process, $\omega_{\alpha}^{\text{Label}_i}$ and $\omega_{\beta}^{\text{Label}_i}$ sets are calculated using the co-occurrence matrix and the rules are created. Exploiting the ontology rules from a huge co-occurrence matrix is a very intensive task, hence the need to distribute the resolution step through the MapReduce paradigm.

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

- key is a tuple $(\text{term}_i, \text{term}_j)$ 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 a alpha rule, or a term comprised in a beta rule

According to the MapReduce paradigm, the pairs are shuffled by key (i.e. $(RuleType, label_i)$) and the 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 $label_i$. The rules are serialized in SWRL language and stored in the ontology-described knowledge base using the OWL-API library.

The generated rules are used in the Realization process to label 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 in Map-reduce 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* approach by using a triple-store with a backward-chaining inference engine. Due to backward-chaining query performance issues identified in [7] 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's RHS (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's LHS (i.e. $(hasTerm(?item, term_i))$ are activated.

The OWL-API library is used to populate the ontology-described knowledge base and a scalable triple-store called Stardog (http://docs.stardog.com) is used to store and manage it. 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 the preliminary results of the proposed classification process are discussed. First the dataset, the environment and the settings used to test the process are described. Then the experiments results are presented and discussed.

5.1 Test environment

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.

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

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 [19] are used as input where the number of terms, labels, and subsumption relations are presented in Table 6.

5.2 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 previous work [19] 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. 2. The reader can observe a decrease in the number of learned 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. 3. 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.

Table 5. Execution Settings

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

Table 6. Previous Results

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

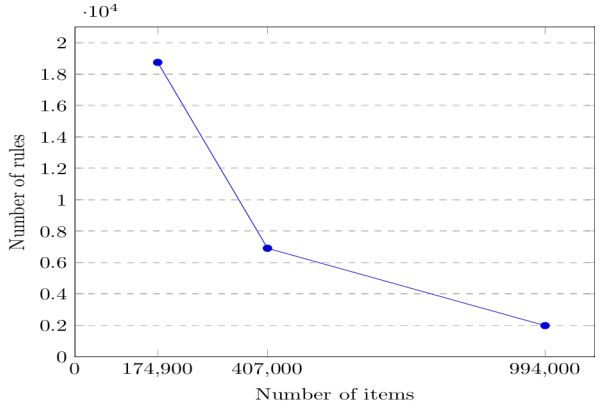

Figure 2. Number of learned rules according to each dataset.

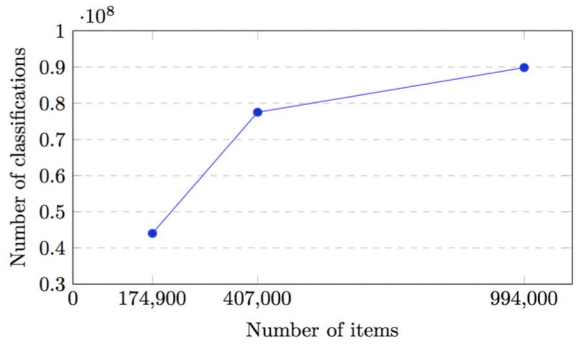

Figure 3. Number of classifications (learned isClassified relations) according to each dataset.

6. CONCLUSIONS

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

First the label hierarchy is automatically obtained and used as the first input for the ontology construction. Then for each label is created a set of rules 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 state of the art 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. First results show that the hierarchy and the enrichment rules are automatically learned and the items classified with the learned labels automatically.

Our current work consists in evaluating the resulting ontology, considering three different aspects: the process scalability (performance), the quality of the hierarchy, and the quality of the classification process (i.e. concept tagging of items).

7. ACKNOWLEDGMENTS

This project is funded by the company Actualis SARL, the French agency ANRT and through the Portuguese COMPETE Program under the project AAL4ALL (QREN13852).

REFERENCES

[1] Ben-David, D. et al. 2010. Enterprise Data Classification Using Semantic Web Technologies. *9th International Semantic Web Conference - Volume Part II* (2010), 66–81.

[2] Bi, W. and Kwok, J. 2011. Multi-label classification on tree- and DAG-structured hierarchies. *Yeast*. (2011), 1–8.

[3] Chen, M. et al. 2014. Big Data: A Survey. *Mobile Networks and Applications*. 19, 2 (Jan. 2014), 171–209.

[4] Dean, J. and Ghemawat, S. 2008. MapReduce : Simplified Data Processing on Large Clusters. *Communications of the ACM*. 51, 1 (2008), 1–13.

[5] Elberrichi, Z. et al. 2012. Medical Documents Classification Based on the Domain Ontology MeSH. *arXiv preprint arXiv:1207.0446*. (2012).

[6] Fang, J. et al. 2010. Documents classification by using ontology reasoning and similarity measure. *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on* (2010), 1535–1539.

[7] De Farias, T. et al. 2015. FOWLA, A Federated Architecture for Ontologies, *RuleML 2015, LNCS 9202* (2015), 97–111.

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

[9] Galinina, A. and Borisov, A. 2013. Knowledge modelling for ontology-based multiattribute classification system. *Applied Information and Communication* (2013), 103–109.

[10] Hassan, T. et al. 2014. Semantic HMC for big data analysis. *Big Data (Big Data), 2014 IEEE International Conference on* (2014), 26–28.

[11] Hitzler, P. and Janowicz, K. 2013. Linked data, big data, and the 4th paradigm. *Semantic Web*. 4, (2013), 233–235.

[12] Horst, H.J. 2005. Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. *Web Semantics: Science, Services and Agents on the World Wide Web*. (2005), 79–115.

[13] Johnson, I. et al. 2010. Making ontology-based knowledge and decision trees interact: an approach to enrich knowledge and increase expert confidence in data-driven models. *Knowledge Science, Engineering and Management*. 304–316.

[14] Kalyanpur, A. et al. 2007. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*.

[15] Möller, R. and Haarslev, V. 2009. Tableau-based Reasoning.

[16] Motik, B. 2009. Hypertableau Reasoning for Description Logics. 36, (2009), 165–228.

[17] Motik, B. and Sattler, U. 2006. A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes. (2006), 227–241.

[18] Obrst, L. 2003. Ontologies for semantically interoperable systems. *Proceedings of the twelfth international conference on Information and knowledge management - CIKM '03* (2003), 366–369.

[19] Peixoto, R. et al. 2015. Semantic HMC: A Predictive Model using Multi-Label Classification For Big Data. *The 9th IEEE International Conference on Big Data Science and Engineering (IEEE BigDataSE-15)* (2015).

[20] Syed, A. et al. 2013. The Future Revolution on Big Data. *Future*. 2, 6 (2013), 2446–2451.

[21] Tsarkov, D. and Horrocks, I. 2006. FaCT++ Description Logic Reasoner: System Description. *Proceedings of the Third International Joint Conference* (2006), 292–297.

[22] Urbani, J. et al. 2010. OWL reasoning with WebPIE: calculating the closure of 100 billion triples. *The Semantic Web: Research and Applications*. Springer. 213–227.

[23] Urbani, J. et al. 2011. QueryPIE: Backward Reasoning for OWL Horst over Very Large Knowledge Bases. *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I* (Berlin, Heidelberg, 2011), 730–745.

[24] Urbani, J. 2013. Three Laws Learned from Web-scale Reasoning. *2013 AAAI Fall Symposium Series* (2013).

[25] Urbani, J. et al. 2012. WebPIE: A Web-scale parallel inference engine using MapReduce. *Web Semantics: Science, Services and Agents on the World Wide Web*. (2012), 59–75.

[26] Vogrincic, S. and Bosnic, Z. 2011. Ontology-based multi-label classification of economic articles. *Comput. Sci. Inf. Syst.* 8, 1 (2011), 101–119.

[27] Werner, D. et al. 2014. Using DL-Reasoner for Hierarchical Multilabel Classification applied to Economical e-News. *Science and Information Conference* (2014), 8.

[28] Wu, H. et al. 2013. A distributed rule execution mechanism based on MapReduce in sematic web reasoning. *5th Asia-Pacific Symposium on Internetware*. (2013), 1–7.