Fine-grained Web Content Classification via Entity-level Analytics: The Case of Semantic Fingerprinting

Govind, Céline Alec and Marc Spaniol

Université de Caen Normandie, Department of Computer Science
Campus Côte de Nacre, F-14032 Caen, France
E-mail: govind@unicaen.fr; celine.alec@unicaen.fr;
marc.spaniol@unicaen.fr

Received October 2018;
Accepted February 2019

Abstract

Approaching three decades of Web contents being created, the amount of heterogeneous data of diverse provenance becomes seemingly overwhelming and its organization is a "continuous battle" against time. In parallel, business, sociological, political, and media analysts require a structured access to these contents in order to conduct their studies. To this end, concise and – at the same time – efficient engineering methods are required to classify Web contents accordingly. However, the whole task is not as simple as classifying something as `A` or `B`, but to assign the most suitable (sub-)category for each Web content based on a fine-grained classification scheme. In practice, the underlying type hierarchies are commonly excerpts of large scale ontologies containing several hundreds or even thousands of (sub-)types decomposed into a few top-level types. Having such a fine-grained type hierarchy, the engineering task of Web content classification becomes out-most challenging. Our main objective in this work is to investigate whether

Journal of Web Engineering, Vol. 17_6&7, 449–482.
doi: 10.13052/jwe1540-9589.17673

entity-level analytics can be utilized to characterize a Web content and align it onto a fine-grained hierarchy.

We hypothesize that "*You know a document by the named entities it contains*". To this end, we present a novel concept, called "Semantic Fingerprinting" that allows Web content classification solely based on the information derived from the named entities contained in a Web document. It encodes the semantic nature of a Web content into a concise vector, namely the semantic fingerprint. Thus, we expect that semantic fingerprints, when utilized in combination with machine learning, will enable a fine-grained classification of Web contents. In order to empirically validate the effectiveness of semantic fingerprinting, we perform a case study on the classification of Wikipedia documents. Even further, we thoroughly examine the results obtained by analyzing the performance of Semantic Fingerprinting with respect to the characteristics of the data set used for the experiments. In addition, we also investigate performance aspects of the engineered approach by discussing the run-time in comparison with its competitor baselines. We observe that the semantic fingerprinting approach outperforms the state-of-the-art baselines as it raises Web contents to the entity-level and captures their core essence. Moreover, our approach achieves a superior run time performance on the test data in comparison to competitors.

Keywords: Fine-grained Web Content Classification, Entity-level Web Analytics, Advanced Web Engineering, Web Semantics, Semantic Fingerprinting.

1 Introduction

Freetext-based keyword search "à la Google" has become a de-facto standard for most Web information systems. However, with the 30st anniversary of the Web approaching and the abundance of Web contents created and consumed 24/7, advanced Web engineering methods are required, as well. For instance, the increased interest of business, sociological, political, and media analysts in conducting their studies on Web documents imposes new challenges on providing more

structured access onto Web data in order to avoid an information-overflow. To this end, methods are required that support fine-grained content classification not only precisely, but also efficiently. This means, in an ideal case, that the Web content's semantics should be fully automatically exploited and interpreted. While artificial intelligence and deep learning [15] have made good progress in text processing recently, semantic analysis of this kind is still in its infancy. Thus, we aim at "understanding" and interpreting a Web content based on its semantic "context", in particular, the named entities it contains. As such, we "judge a document based on the named entities it contains", which is an adaptation of Firth's famous statement: "You shall know a word by the company it keeps!" [5] in the direction of entity-level analytics.

In order to understand a Web content, a human requires to put the document in its context. To this end, the "crucial" information are identified, aggregated and interpreted. Key-point in this process is the identification and contextualization of the named entities contained that allow us to raise Web contents from "strings" to "things" [8]. Our research hypothesis therefore is: named entities contained in a Web content are "type-specific" and characteristic. We postulate that Web contents to be classified, e.g., as type `(football-)club`, should not only be classified based on the "terms" contained, but more appropriately by the "things", e.g., `(football-)players` and `(football-)stadiums`. For that purpose, we harvest entity information and aggregate them as so-called "semantic fingerprints". These fingerprints then allow us to efficiently classify Web contents.

In this paper, we introduce "The Case of Semantic Fingerprinting" as a novel approach toward fine-grained entity-level content classification. Our goal is to align Web contents with respect to a fine-grained type hierarchy as depicted in Figure 1. This task is inherently complex as it involves finding the most closely related type to the Web content from a taxonomy with large number of types and multitude of hierarchical relations between them. In particular, we investigate the classification of Wikipedia articles based on their "inherent semantics" derived from the YAGO knowledge base [7, 23]. This implies the exploitation and distillation of the entity-related information in Web contents for a subsequent

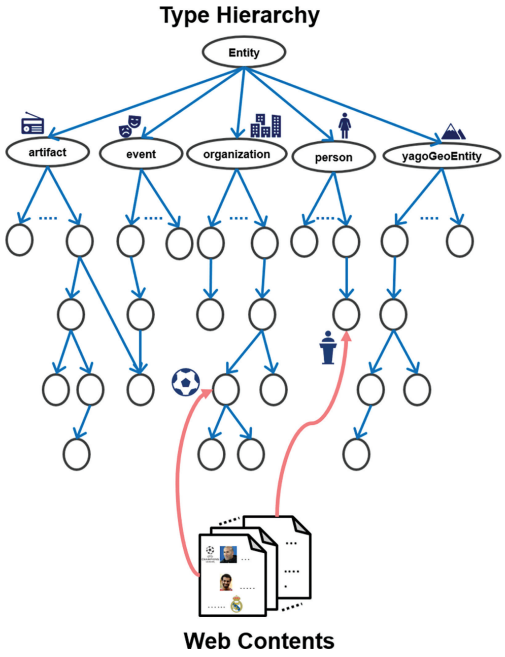


Figure 1 An illustration on the alignment of Web contents onto a fine-grained type hierarchy.

classification according to a fine-grained classification scheme. To this end, we formulate the key hypothesis behind this work as follows:

> "*A document can be characterized by the named entities it contains*".

Therefore, we postulate the following:

- **Hypothesis 1:** The semantics of a Web content can be captured via the named entities contained.
- **Hypothesis 2:** A concise and quality representation, i.e., "semantic fingerprint" of Web contents can be obtained by exploiting the entity-level type information.
- **Hypothesis 3:** "Semantic Fingerprints" when backed by machine learning can provide a sophisticated fine-grained type classification model for Web contents.

In summary, this paper makes the following salient contributions: firstly, we present a novel concept, called "Semantic Fingerprinting" that allows Web content classification solely based on the information derived from the named entities contained in a Web document, outperforming state-of-the-art methods. Secondly, we thoroughly examine the results obtained by giving in-depth insights about the performance of Semantic Fingerprinting with respect to an analysis of a wide range of parameters inherent in the data. Finally, we conduct experiments on performance aspects of the developed approach in order to show the viability of our method.

2 Related Work

Type classification has been studied on various levels of granularity. In the following subsections, we segregate and discuss prior research based on entity and document level of granularity, i.e., the entity type classification, and the document type classification.

2.1 Entity Type Classification

Assigning the most appropriate type(s) to individual entities is a crucial fundamental task, and is addressed by entity type classification. In [6], the authors propose a supervised method to determine the fine-grained types of entities. They consider the local contextual information as well as the global knowledge derived from resources such as WordNet [20]. Using the combination of before mentioned features, they train multiple classification models such as k-NN (k-Nearest Neighbors), NB (Naive Bayes), SVM (Support Vector Machines), and C4.5 decision tree. The decision tree model is reported to be performing the best.

FIGER (FIne-Grained Entity Recognition) is a fine-grained entity recognizer, which adapts the perceptron classifier to predict the fine-grained entity tags derived from Freebase [17]. This work employs a variety of features, which include contextual n-grams, PoS (Part of Speech) tags, syntactic dependency, and distributional similarity features. A CRF (Conditional Random Fields) based model is utilized for the segmentation, and for the assignment of types to the named entities a multi-class multi-label perception based classifier is adapted.

They perform an extrinsic evaluation on the relation extraction task to measure the performance of their approach.

In [21], a collective and hierarchical classification model is presented in order to determine the fine-grained semantic classes of nouns (total around 100). In this work, the authors make use of factor graphs to perform collective classification, and more than 30 features of disparate types such as morphological, semantic, grammatical, and gazetteers, etc. They introduce a collective classification method based on factor graphs, which enables to exploit the relational information.

HYENA (Hierarchical tYpe classification for Entity NAmes) is an entity type classification system on a very fine-grained type taxonomy [25]. Here, the authors present a multi-label hierarchical classifier in combination with a meta classifier to predict very fine-grained types. The feature set includes features from context, grammatical structure, gazetteer, etc. It is worth noticing that HYENA does not use features from WordNet in contrast to the system proposed by [21].

However, all of these approaches address type classification of individual entities in contrast to type classification of the overall document containing them. Hence, these approaches are related but should not be considered as directly comparable to our work.

2.2 Document Type Classification

In one of the earlier works on document type classification, the use of various supervised machine learning models is explored in [22]. The author examines various aspects of the document classification problem and survey the performance of several machine learning based classification techniques.

Furthermore, in [10], a support vector machine classifier is employed to learn with many relevant features. They represent a document as a tf-idf (term frequency - inverse document frequency) vector. A support vector machine classifier based on the RBF (Radial Basis Function) kernel is utilized to perform the categorization.

In [4], the utilization of concepts from the WordNet ontology is investigated in combination with terms in documents to aid the classification task. A vector comprised of tf-idf features along with

concept features from WordNet represents the document. They perform chi-square feature selection and employ a Naive Bayes classifier to categorize text documents.

In [12], the authors present a fast and simple text classification technique based on a linear classification model with rank constraint. The authors utilize only the n-gram features to build the model. As there can be huge number of n-grams, a hashing trick is employed to access features faster. In this work, they also make use of hierarchical softmax to efficiently deal with the higher number of output classes.

An ontology-based text classification method [2] aims at classifying documents with respect to dynamically defined topics. The authors claim that their model does not require a predefined set of output classes and the corresponding annotated training set. The model can adapt to any given ontology of output labels, as it learns to map the thematic sub-graphs created from documents to the given ontology.

[16] experiments combining the tf-idf features with word2vec [19] word embeddings. They present a classification model based on SVM that uses word2vec features weighted by tf-idf along with the original tf-idf features itself. The authors report that the model trained with this combination outperforms the models based upon individual set of features.

[1] uses an ontology to associate documents describing an entity with their types. To do that, they automatically populate a domain ontology with respect to the information contained in each document. They use an ontology-based machine learning tool to learn a definition for each type. If a document complies with the definition of a certain type, then it is classified as this type.

Recent approaches employ DNNs (Deep Neural Networks) that perform better when given a vast amount of training data. Examples of such studies include, utilizing the word order of the textual data for document classification [11]. In this work, the authors adapt a CNN (convolutional neural network), which can operate on the input text of variable size. The proposed CNN model utilizes the word ordering information in the text to predict the document type.

A hierarchical attention network based on GRU (Gated Recurrent Unit) is presented by [24], which captures the hierarchical nature of

documents. In this work, a two level attention mechanism is introduced, which operates at word and sentence level of a document. These two attention mechanisms focus on finding the important parts of documents to build a better representation that can be further utilized for classification.

In [13], a text classification model based on the convolutional neural network is proposed, which has been reported to provide encouraging results on a variety of classification tasks (such as sentiment analysis, ratings prediction, etc.). Here, the author makes use of word2vec pre-trained word embeddings, and presents several variants of static and dynamic word representation.

A classification system that employs both the RNN (Recurrent Neural Network) as well as the CNN is introduced in [14]. The proposed model has a convolutional module, which recurrently reads the sequence of words to obtain a document representation of higher quality. In addition, the authors introduce a max-pooling layer in the network that helps in determining which of the words in the text are of more importance to achieve the overall goal with higher accuracy.

In contrast to these works, our approach exploits entity-level semantics to build a concise document representation (i.e., the semantic fingerprint), and does not have dependency on huge training data. In addition, we observed that there are not many document level type classification studies that target a large number of output classes. In this work, we explore the task of Web content classification with respect to a fine-grained hierarchy (with over 100 types).

3 Computational Model

"Google-style" indexing based on (freetext) keywords has been widely understood and adopted by the public audience. With the maturity of the Web and its quality as a primary field of study, e.g., in the context of "Web Science" [3], there has been recently a trend towards semantically classifying Web contents. With the emergence of large scale and fine-grained classification systems available via the LOD cloud, there is further an increased demand for interlinking (or simply classifying contents) based on an underlying ontology, e.g., the Wikipedia category system.

Typically, a category prediction method involves collection of the document features followed by the application of a machine learning model such as logistic regression, Naive Bayes, etc. The most commonly utilized features for this purpose are bag-of-words. These representations are usually sparse in nature, and hardly captures the semantics of the document. We propose semantic fingerprinting, a method to represent the text documents based on the named entities it contains. This method exploits the type characteristics of the involved entities and produces a concise representative vector for the Web content by capturing its semantic essence. We present our complete approach in the following subsections.

3.1 Type Hierarchies & Semantic Content Classification

The task of semantically classifying Web contents aims at assigning the proper type(s)/category(ies) associated. To this end, we utilize a fine-grained type hierarchy extracted from the YAGO knowledge base [23, 7, 26]. This type-system is based on 5 top-level types (`person`, `location`, `organization`, `event` and `artifact`). In particular, we have chosen the 20 most populated (sub-)types per top-level type in order to automatically build our classification system based on "popularity". It is worth mentioning, that the resulting structure is a directed acyclic graph (DAG) and not a tree. It implies that certain (sub-)types might be associated with more than one super-type in order to express a context-depending facet of this type. This leads to a hierarchy consisting of 105 types. A graphical representation regarding details on the structure and types contained in the type hierarchy is available on the project website[1]) as well as in Appendix A1.

3.2 Type Score Vectors

In order to "semantically" summarize a document, we exploit the set E of entities contained, where each entity e_i belongs to a set of types denoted by $types(e_i)$. Let T be the set of all possible types, and the hierarchy H defining the relationship among them. For an entity e_i, the

[1] https://spaniol.users.greyc.fr/research/Semantic_Fingerprinting/105_hierarchy.pdf

score of each of the types in T is aggregated by hierarchical upward propagation of the values of each type t in $types(e_i)$.

$$score(t) = score(t) + \sum_{k \in H(t)} score(k) \tag{1}$$

The children of a type t are given by $H(t)$. Using the recursive process given in Equation 1, a type score vector is computed for each entity e_i, which contains an aggregated score entry for each of the types in T. A semantic fingerprint $d \in D$ is the vector representation of the document, and is defined in the following subsection. Figure 2 illustrates the computation of the type score vector for the entity `Real Madrid` via a small part of our type hierarchy. The entity `Real Madrid` is directly associated with the types `team` and `club`, which are discovered using a knowledge base. To compute the type score vector, we assign a score of 1 to the nodes in our hierarchy representing types `team` and `club`. All other nodes are initialized with a score of 0. Now, we propagate the scores of children nodes to the parent nodes, and perform score aggregation as aforementioned. We keep on repeating the process until we reach the root node in the type taxonomy. The final scores are represented by a vector as depicted in Figure 2.

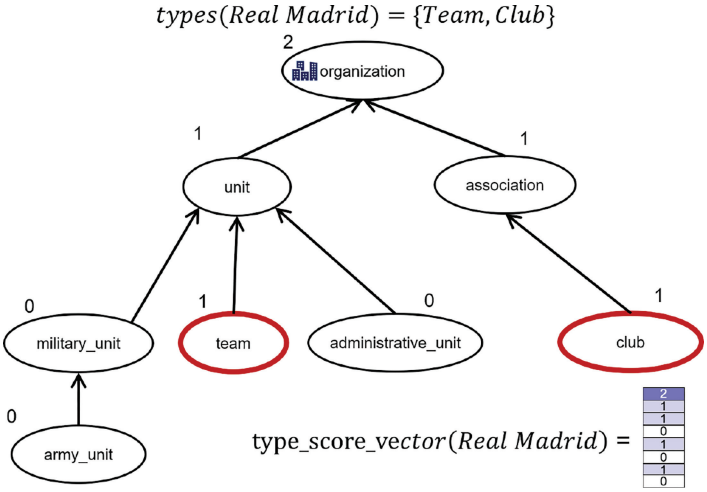


Figure 2 An example on a small fragment of our type hierarchy illustrating the computation of a type score vector for an entity.

3.3 Semantic Fingerprint

As a semantic fingerprint, we define the vector d for a document, which is computed by summing the individual type score vectors for all the associated named entities $e_i \in E$ in the document/Web content:

$$\begin{bmatrix} d_1 \\ d_2 \\ . \\ . \\ d_{|T|} \end{bmatrix} = \underbrace{\begin{bmatrix} t_1 \\ t_2 \\ . \\ . \\ t_{|T|} \end{bmatrix}}_{e_1} + \underbrace{\begin{bmatrix} t_1 \\ t_2 \\ . \\ . \\ t_{|T|} \end{bmatrix}}_{e_2} + \cdots + \underbrace{\begin{bmatrix} t_1 \\ t_2 \\ . \\ . \\ t_{|T|} \end{bmatrix}}_{e_{|E|}}$$

Figure 3 illustrates the generic process of computation of the semantic fingerprint for a document. As depicted, the process start with the computation of a type score vector for each of the individual named entities. The type score vector is of the same dimension as the semantic fingerprint, and is computed as discussed in previous subsection.

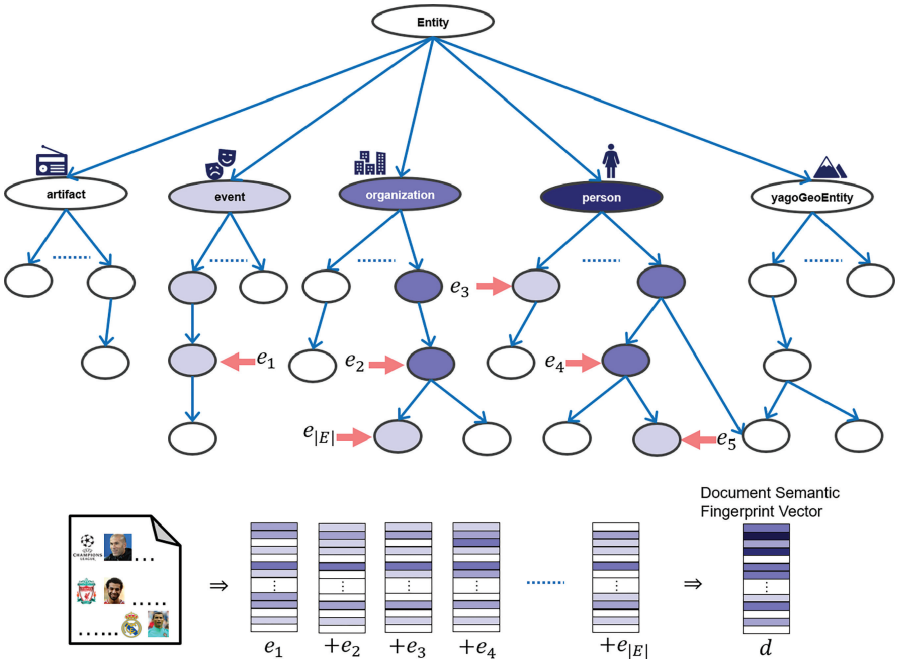


Figure 3 An illustration of the computation of semantic fingerprint for a document.

The color intensity in the figure, is representing the magnitude of values. The document semantic fingerprint vector d is the summation of type score vectors. Figure 3 also depicts a sample of the type hierarchy with nodes colored with different intensities depending on their scores resulting from the aggregated type score vectors. It captures the intuition of how the sum over type score vectors aggregates the overall semantics of a document.

4 Content Classification via Semantic Fingerprinting

The originality of our approach is the use of semantic fingerprints that allows us to predict one or more fine-grained types to be associated with a given document. What we call a semantic fingerprint is a vector representing the entity types contained in a document. As such, we raise analytics to the entity-level, which has several advantages: firstly, it is compact as it consists of 105 types, only. As a result, the corresponding vector(s) are by orders of magnitude smaller than "bag-of-words" representations of the same content and, thus, more efficient to process. Finally, utilizing entity-level information means the exploitation of semantics. To this end, "Semantic Fingerprinting" is based on the following four consecutive steps. Figure 4 depicts the conceptual approach graphically.

(1) Named Entity Recognition and Disambiguation
For each document to be classified, we extract and disambiguate the named entities $e_i \in E$. Conceptually, it is done, e.g., by AIDA [9] to disambiguate onto YAGO. In the case of Wikipedia, we obtain the entities directly via the mark-up. If the same named entity is present more than once in the content, its multiplicity does not count to the system as we consider only the unique entities contained.

(2) Entity Type Hierarchy Computation
For each named entity e_i contained in a document, we derive the associated types from the underlying type hierarchy. In original, the named entities are labeled with as per type hierarchy of YAGO. As YAGO has huge type hierarchy, we map the type system used in YAGO

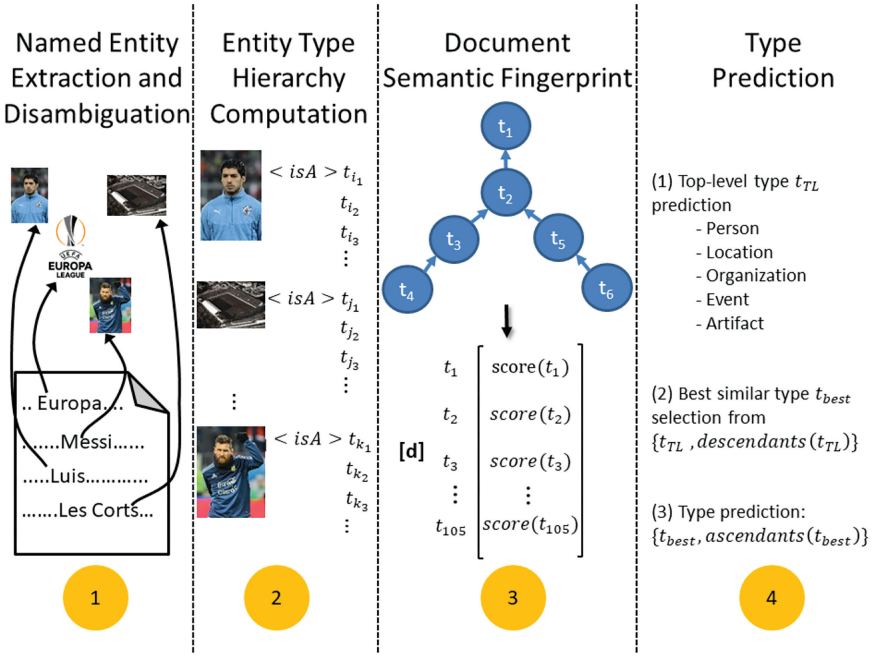


Figure 4 Conceptual approach by the example of a document of type `club`.

to the type hierarchy considered here to filter out the infrequent types. By doing so, we are able to identify all associated (sub-)types out of the 105 before mentioned types (cf. Section 3.1).

(3) Document Semantic Fingerprint

The semantic fingerprint of a document is represented by the vector consisting of an entry for each of the 105 types. Each value is the aggregated score for the entities of this type contained in the document. As discussed before the idea is to capture the core semantics behind the document and represent it by a vector of fixed length.

(4) Type Prediction

Type prediction based on the semantic fingerprints requires the computation of a representative vector for each type $t \in T$. In order to do so, we construct for each of the 105 types a representative vector aggregated

from 100 randomly drawn documents per type. Each document is processed as described in steps 1–3 to compute the corresponding semantic fingerprint.

For the actual type prediction, we utilize a two-step process. This two step prediction process is needed to first identify the "characteristic pattern" of a type and then predict its "specific (sub-)type". First we use a classifier to learn and predict the top-level type. As mentioned in Section 3.1 the resulting type system is a DAG, so that (sub-)types with multi-parent nodes exist. Thus, a document can be typed by several top-level types. In this case, we randomly assign one of its top-level types in the training and the test sets. Once we have predicted the top-level type, we compute the cosine similarity as shown in Equation 2 between the representative vector of the document and the representative vector of each type that is a descendant of previously selected top-level type. We then select the one with the highest score and all its ascendant types.

$$cosine(X,Y) = \frac{\sum_{i=1}^{|T|} X_i * Y_i}{\sqrt{\sum_{i=1}^{|T|} X_i^2}\sqrt{\sum_{i=1}^{|T|} Y_i^2}} \tag{2}$$

5 Empirical Validation and Case Study on Wikipedia

In this section we evaluate our method of "Semantic Fingerprinting" (indicated by $SemanticFingerprint$) against two baseline methods "Naïve Bayes" [18] and "Elberrichi" [4] (referred by $NaiveBayes$ respectively $Elberrichi$). "Naïve Bayes" is a common baseline used in document classification. In this experiment, it is used with bag-of-words vectors composed of the frequency of the words. The second baseline, "Elberrichi", is a "semantic competitor" employing WordNet [20] on the most relevant noun phrases.

5.1 Evaluation Data Set

We use Wikipedia as the source of ground truth. In order to do so, we have chosen 10,500 random examples for training and 1,050 for testing. As Naïve Bayes benefits from prior probabilities, we maintained the ratio in the number of training examples accordingly. For the other two

approaches, the training set consists of 100 random examples for each of the 105 types. The test set comprises 10 random examples for each type. Markup and stop words are removed during the preprocessing. In our $SemanticFingerprint$ method, a random forest classifier is employed to select a single top-level type. It is trained with the full training set in each bag and computing the attribute importance with mean impurity decrease. This classifier achieves accuracy of around 68% on the top-level types.

5.2 Evaluation Strategy

We evaluate the performance of our approach with respect to Precision, Recall and F1 (cf. Equations 3 to 5). Here, TP, FP, FN and FP correspond to true positives, false positives, false negatives, and false positives respectively. Figure 5 highlights various examples for type 'C' prediction. Example 1 is actually more specific (it is an example of 'D'). Example 2 is the same as 1 but it is also multi-labeled: it is typed as both 'D' (and its ascendants) and 'E' (and its ascendants). In both cases, we consider a prediction leading to the type 'F' (and its ascendants). In our data set, most cases are like Example 1 (only a few of them ($\sim 10\%$) are multi-labeled). Thus, we do not apply multi-label prediction. In the following, we explain two different evaluation concepts that enable the assessment of various aspects of hierarchical multi-labeled data.

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{5}$$

Full Evaluation

In case of the full evaluation, we consider all types annotated to a test example in the ground truth and all the predicted types by concerned approach. We provide an illustration for the evaluation methods in

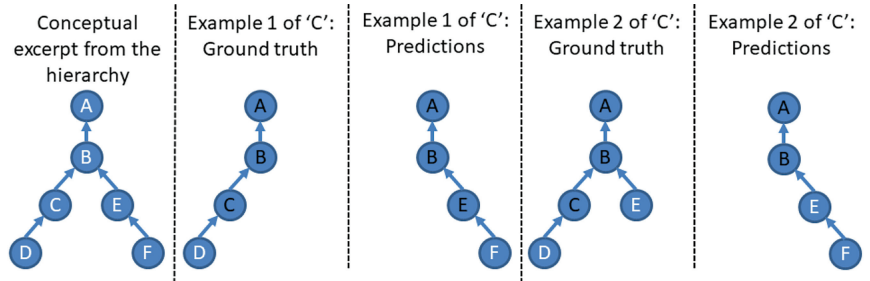


Figure 5 Full and focused evaluation.

Figure 5. For Example 1, types denoted by 'A' and 'B' are true positives, whereas the types 'A', 'B' and 'E', are true positives for Example 2.

Focused Evaluation

In this setting we predict an exact type. To this end, we consider as ground truth only this specific type and its ascendants ('C' and the nodes with black label in Figure 5) as ground truth. Thus, we assess all predictions against all types up to the level of the assessed node: level 3 for node 'C' in our examples, so we consider nodes on that level or above from the predictions. Moreover, in case of ground truth with multi-labeled types (such as Example 2), we remove from the predictions the correctly predicted types that are not the focus of the ground truth. In particular, in Example 2 'E' (level 3) is removed from the predictions because it is not considered as ground truth anymore, since the assessment is focused on 'C' and its ascendants.

5.3 Experimental Results

We report the macro-averaged as well as micro-averaged results, and the evaluation over the test set is summarized in Table 1, respectively Table 2. Equations 6 to 9 define the used macro-averaged and micro-averaged measures where N is number of examples in the test set and subscript i is used to denote the measures related to some i^{th} example in the test set.

Table 1 Macro-average scores

	Full Evaluation			Focused Evaluation		
Method	Precision	Recall	F1	Precision	Recall	F1
$NaiveBayes$	0.52032	0.12026	0.19537	0.51651	0.19588	0.28404
$Elberrichi$	0.52421	0.42322	0.46833	0.53853	0.48325	0.50940
$SemanticFingerprint$	**0.59992**	**0.43766**	**0.50610**	**0.62063**	**0.52097**	**0.56645**

Table 2 Micro-average scores

	Full Evaluation			Focused Evaluation		
Method	Precision	Recall	F1	Precision	Recall	F1
$NaiveBayes$	0.55185	0.10270	0.17317	0.54626	0.15866	0.24590
$Elberrichi$	0.56854	0.41848	0.48210	0.58719	0.50207	0.54130
$SemanticFingerprint$	**0.62545**	**0.42845**	**0.50854**	**0.63756**	**0.51318**	**0.56865**

$$macro\text{-}precision = \frac{\sum_i precision_i}{N} \tag{6}$$

$$macro\text{-}recall = \frac{\sum_i recall_i}{N} \tag{7}$$

$$micro\text{-}precision = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i} \tag{8}$$

$$micro\text{-}recall = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i} \tag{9}$$

Our method ($SemanticFingerprint$) outperforms the other approaches in all performance measures and evaluation strategies. $NaiveBayes$ performs the weakest, due to the complexity of the prediction problem and the lack of clearly discriminating features on the word-level. The second "semantic" method incorporating WordNet introduced by $Elberrichi$ performs slightly better than $NaiveBayes$, but is still more than 5% weaker than our approach. In this setting, $Elberrichi$ uses 200 features to represent a type as it is claimed to be best configuration by the authors. Thus, $Elberrichi$ requires almost double the number of features as compared to $SematicFingerprint$. In addition, it employs both the terms as well as the concept frequencies in feature set and requires a computationally expensive feature selection step. $SematicFingerprint$ uses feature vectors of size 105 to represent types and documents. Nevertheless, it gains over $Elberrichi$

around 3.8% and 5.7% in macro-F1 for the full evaluation and the focused evaluation as well as around 2.6% and 2.7% in micro-F1. In summary, $SematicFingerprint$ benefits from the compact and concise representation incorporating information derived from entity-level, which proves our Hypothesis 1-2 as postulated in Section 1. Furthermore, the beneficial utilization of machine learning classifier (i.e., RF with semantic fingerprints as features) for the top-level type prediction has also proven the Hypothesis 3 to be viable. All data used in our experiments can be found in the associated zip-file[2].

We analyzed the performance of our method and baselines at the top-level in the type hierarchy, i.e., with the 5 top-level types (`yagoGeoEntity`, `person`, `organization`, `artifact`, and `event`). The top-level types scores for the 3 methods are reported in Table 3. Our $SemanticFingerprint$ approach has outperformed both the baselines with achieving best F1-score for 4 out of the 5 types. Naive Bayes performs better in F1 than the other two methods for `yagoGeoEntity` type. Each of the methods perform their personal best on `yagoGeoEntity` type due to the fact that these pages tend to be more homogeneous as compared to the other groups. Moreover, pages belonging to `yagoGeoEntity` stand out because they are some of the earliest created Web pages in the dataset (cf. Section 5.5). $NaiveBayes$ has the lowest performance for `organization` type and $Elberrichi$ struggles with the pages of type `person`. On the contrary, as hypothesized in Section 1, by learning the discriminative patterns via semantic fingerprinting representation of Web contents our approach provides good performance over all types.

Table 3 Top-level type specific scores

	$NaiveBayes$			$Elberrichi$			$SemanticFingerprint$		
Type	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
`yagoGeoEntity`	0.707	0.948	**0.810**	0.739	0.797	0.767	0.738	0.814	0.774
`person`	0.349	0.995	0.517	0.531	0.362	0.431	0.679	0.705	**0.692**
`organization`	1.000	0.086	0.158	0.496	0.619	0.551	0.676	0.667	**0.671**
`artifact`	0.606	0.500	0.548	0.577	0.579	0.578	0.698	0.597	**0.644**
`event`	0.900	0.114	0.202	0.565	0.570	0.567	0.662	0.696	**0.679**

[2]https://spaniol.users.greyc.fr/research/Semantic_Fingerprinting/data.zip

5.4 Type-specific Performance Analysis

Table 4 shows the best and worst performing types and their corresponding scores for each of the aforementioned approaches. For this purpose, we computed the macro/micro averaged F1-score over the test examples (i.e., 10 examples for each type) in the test set. We report the macro averaged scores for focused evaluation here, because we consider focused evaluation as "purely" type-specific. Moreover, we consider the macro averaged score to be more suitable in this case as it gives equal weightage to each example, and omit the micro averaged one. Apart from that, both follow a similar pattern. We observe that the worst 10 performing types for our $SemanticFingerprint$ approach have better scores than the worst performing types of the baseline approaches. Moreover, the best predictions by the baseline approaches are restricted to clusters of a few types only. This observation can be captured from the color-encoded visualization of the hierarchical type structure as seen in Appendix A2. $NaiveBayes$ fails to achieve good performance on more specific types at the lower levels of the

Table 4 Best and worst performing types for the individual approaches

	$NaiveBayes$		$Elberrichi$		$SemanticFingerprint$	
	Type	Mac-F1	Type	Mac-F1	Type	Mac-F1
Best Performing Types	yagoGeoEntity	1.000	yagoGeoEntity	1.000	person	1.000
	person	1.000	organization	1.000	artifact	1.000
	organization	1.000	artifact	1.000	organization	0.974
	artifact	1.000	event	1.000	event	0.974
	event	1.000	region	1.000	yagoGeoEntity	0.947
	location	0.824	military_unit	1.000	vote	0.937
	instrumentality	0.710	vote	1.000	school	0.933
	geological_formation	0.667	person	0.974	album	0.933
	body_of_water	0.667	container	0.966	medium	0.909
	contestant	0.667	location	0.950	conflict	0.897
Worst Performing Types	computer_game	0.310	magazine	0.373	settlement	0.445
	wheeled_vehicle	0.286	change	0.362	district	0.442
	senior_high_school	0.286	publication	0.360	municipality	0.439
	book	0.286	writer	0.358	magazine	0.389
	beginning	0.286	company	0.353	game	0.387
	motor_vehicle	0.261	book	0.339	press	0.373
	movie	0.250	musician	0.308	alumnus	0.358
	introduction	0.250	manufacturer	0.296	tract	0.353
	city	0.235	alumnus	0.289	diversion	0.350
	magazine	0.182	craft	0.255	village	0.244

hierarchy, while $Elberrichi$ performs strongly on the subtypes of the `yagoGeoEntity` branch but struggles in case of `events` and `persons`. However, $SemanticFingerprint$ maintains the balance among all types and achieves good overall performance across the 5 branches in our fine-grained type hierarchy (cf. Appendix A2).

5.5 Dataset Analysis

In order to better understand the nature of Web contents for the classification task, we conduct a detailed study on our dataset. As named entities are a central component of our method, we investigate their distribution in our dataset used for training and testing, first. To this end, Table 5 shows various statistics about the distribution of entities in Web pages, such as mean, median, mode, standard deviation (SD), etc. We observe that the average number of entities contained is 105 entities, but half of the population has less than 53 entities, and the most probable case is with 14 entities. The big differences between mean and median values are due to the fact that a few pages contain a large number of entities and tend to increase the mean number of entities. However, many pages contain only a few entities, as shown in Figure 6a–6f, the distribution is right skewed and follows a power law. We noticed that Web pages belonging to `person` are more concentrated towards the left, and have the lowest mean and standard deviation as visible in Figure 6b. Moreover, it is worth pointing out that there exist also a few pages belonging to all types except `yagoGeoEntity` that do not contain any entity at all. These pages with very few entities (or even without any) have only little (respectively no) benefit for our method. In total,

Table 5 Statistics about entities in the dataset used for training and testing

Type	Mean	Median	Mode	SD	Min	Max	#Entities
`yagoGeoEntity`	121.30	71.0	14.0	150.54	1	1204	254,746
`person`	70.58	35.5	14.0	112.44	0	2042	148,219
`organization`	115.26	59.0	8.0	151.80	0	1334	242,056
`artifact`	94.55	47.0	27.0	124.55	0	1480	245,836
`event`	125.40	69.0	19.0	168.34	0	3046	275,957
All	105.05	53.0	14.0	144.08	0	3046	1,103,069

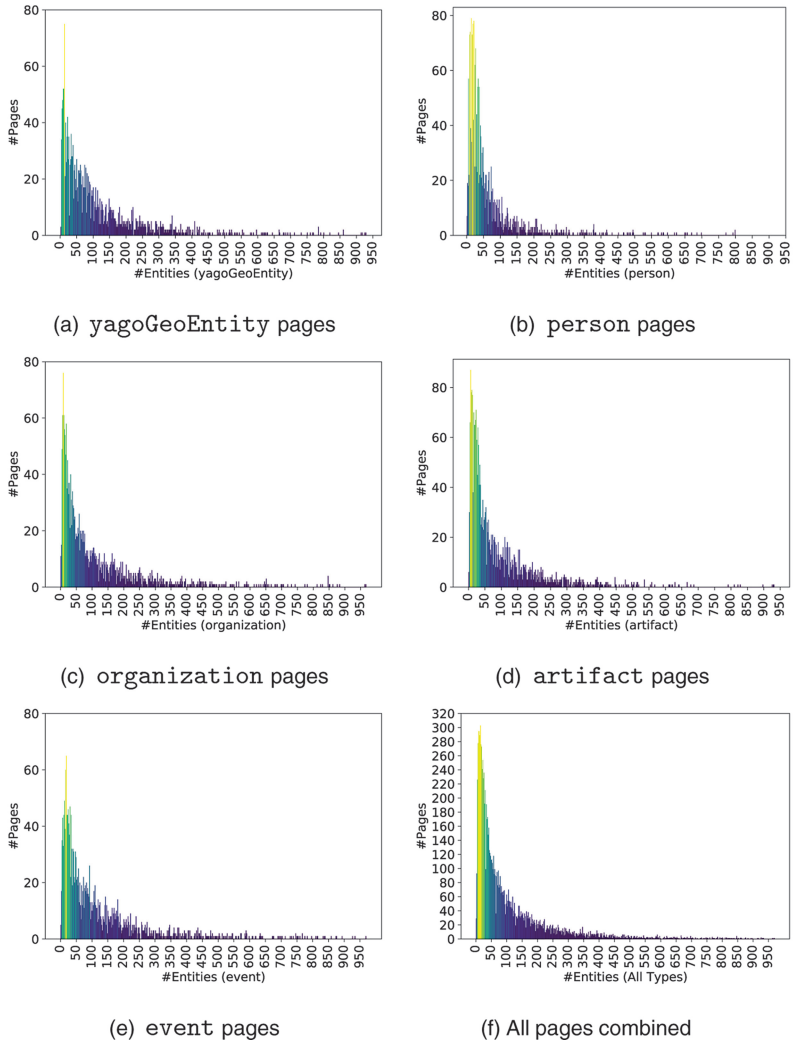


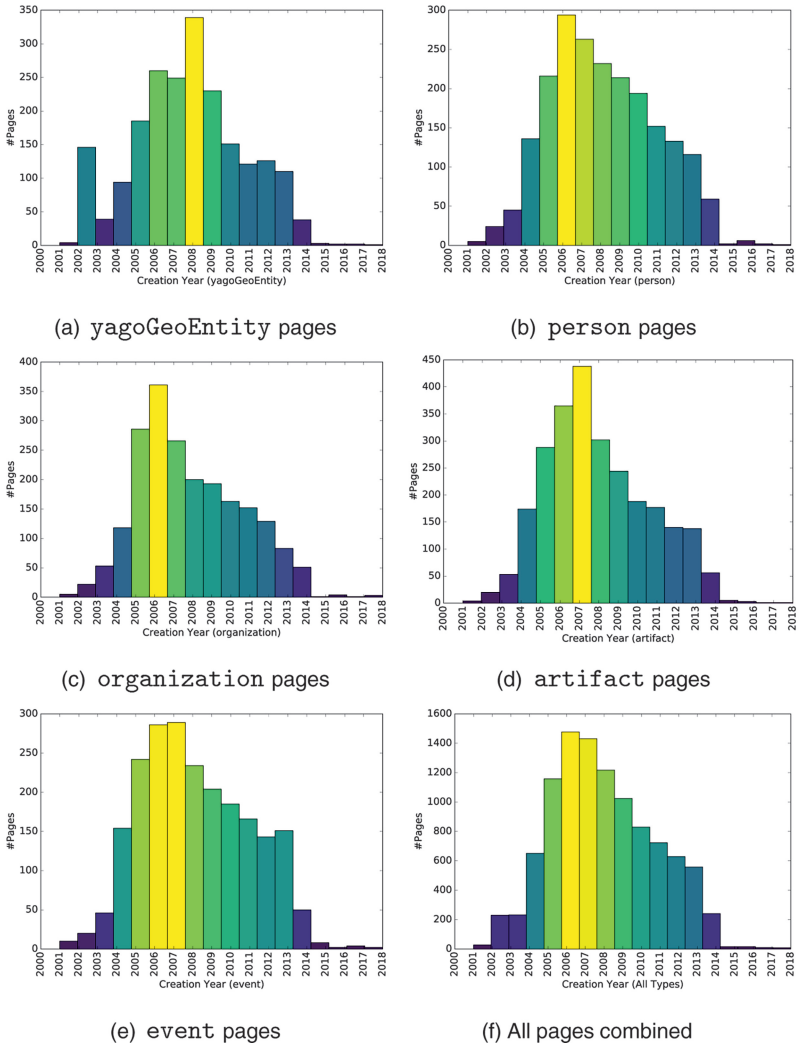
Figure 6 Plots depicting entity counts in dataset pages.

there are 24 pages that have zero entities and another 348 having less than or equal to 5 entities contained.

Concerning the part of our dataset used for testing, there are around 30 pages having less than or equal to 5 entities contained, and we observe the minimum number of entities to be 2. These pages, are

particularly "problematic" for our method due to the fact, that those contents will not generate a characteristic semantic fingerprint. As such, those pages are "defaults by definition" for $SemanticFingerprint$.

Further, we studied the age of Web contents in our dataset. Figure 7f depicts the distribution of the creation of Web pages over years



(a) `yagoGeoEntity` pages (b) `person` pages

(c) `organization` pages (d) `artifact` pages

(e) `event` pages (f) All pages combined

Figure 7 Histograms depicting age distribution of pages in the dataset.

via histograms. As the sample is randomly drawn from Wikipedia, it reflects some of Wikipedia's inherited characteristics. It can be observed from Figure 7a that `yagoGeoEntity` pages are some of the earliest ones that have been created, which also captures the intuition created contents about "'static" concepts first. Further, we notice a high number of overall page creations around the year 2007 (at a time Wikipedia become popular among average users) as it can be seen in Figure 7a–e. Moreover, age distributions of the pages belonging to types `organization` (cf. Figure 7c) and `artifact` (cf. Figure 7d) have lower dispersion over years as compared to types `event` and `person` and overall. The before mentioned effect can also be interpreted as a continuous flow of pages about these types being added.

5.6 Wall-clock Time Comparison

We perform experiments to analyze the wall-clock time consumption for the different approaches presented in this paper. We report the training and the testing time of individual approaches in Table 6 by averaging over multiple runs in the same computational environment. All experiments are conducted on a Linux machine with Intel Core i5-4690 CPU (clock rate 3.50GHz) and 8GB of randomly accessible physical memory. The required wall-clock time for each approach is intentionally recorded by excluding the preprocessing steps (such as lemmatization, named entity disambiguation, etc.), in order to allow a fair assessment of the plain categorization process. This is indispensable for obtaining comparable results, because the individual preprocessing steps are not core part of approaches and their computational efficiency

Table 6 Wall-clock times of the presented approaches (in milliseconds)

Method	Train time	Test time (Focused)	Test time (Full)
$NaiveBayes$	**10375.2**	4183.0	3922.2
$Elberrichi$	79338.8	2014.8	1946.0
$SemanticFingerprint$	32183.4	**1085.2**	**760.4**

can vary with respect to the tools employed. We report the wall-clock time required to perform the evaluation on whole the test set in correspondence to the aforementioned evaluation approaches (focused and full). It can be observed from Table 6 that $SemanticFingerprint$ outperforms its competitors in computation time required for classification. In training, it performs second best. The excellent performance in classification is attributed to the concise representation of document semantics with the help of semantic fingerprints. It is further worth mentioning that the "loss" in training time is somewhat "negligible", since this is a one time procedure, while the actual classification is executed for each and every document to be classified. As such, $NaiveBayes$ has to compute prior and likelihood values, which do not involve the same level of complexity in comparison with both other approaches, and thus, has the fastest training process. Our $SemanticFingerprint$ approach, however, reduces the required training time by more than 50% with respect to $Elberrichi$. $NaiveBayes$ utilizes a large set of terms as features, and $Elberrichi$ employs 200 best features (terms and concepts) selected via chi-square statistics, whereas our $SemanticFingerprint$ approach only requires a 105 dimensional vector for documents and types representation. As a result, this leads to a better execution time performance of $SemanticFingerprint$.

6 Conclusion and Future Work

30 years of the Web require advanced methods of Web engineering. In this paper, we have introduced "The Case of Semantic Fingerprinting" as a novel method for fine-grained content classification based on entity-level analytics. Our study on Web content classification has proven that a document can be characterized by the named entities it contains, as hypothesized initially. By raising contents to the entity-level, we are able to capture the semantics concisely and efficiently. Based on extensive experiments on Web contents classified in Wikipedia, we have shown the viability of our approach and its performance gain against state-of-the-art competitors. The type information related to the named entities contained in a Web content have proven to be

key in determining category of the Web content. As such, "Semantic fingerprints" backed by machine learning can serve as a concise as well effective solution for fine-grained Web content classification at large scale. Nevertheless, types such as `football player` and `club` are not necessarily clearly distinguishable when seen from the contained named entities perspective, as their semantic fingerprints are comparatively similar. To this end, more sophisticated strategies need to be developed.

The encouraging results presented in the experimental section pave the way to further enhancements. As such, upcoming next steps include – but are not limited to – research on even more fine-grained classifications, usage of Semantic Fingerprinting in domain-specific applications as well as improvement of particularly hard to distinguish entity types. Apart from incorporating additional features, we also plan to use features extracted from entities discovered via type-specific entity relations (e.g., the relation "bornIn" for a `person`) and to combine Semantic Fingerprinting with deep learning. For that purpose, we intend to apply RNN (Recurrent Neural Network) in the fine-grained classification process based on the semantic fingerprints.

Acknowledgements

This work was supported by the RIN RECHERCHE Normandie Digitale research project ASTURIAS contract no. 18E01661. We thank our colleagues for the inspiring discussions.

Appendix

A1 Type Hierarchy

The five different branches (i.e., `yagoGeoEntity`, `person`, `organization`, `artifact`, and `event`) of our fine-grained type hierarchy are depicted in Figure A1–A5. Nodes with gray color represent that they either have more than one parent within a branch or belong to multiple branches.

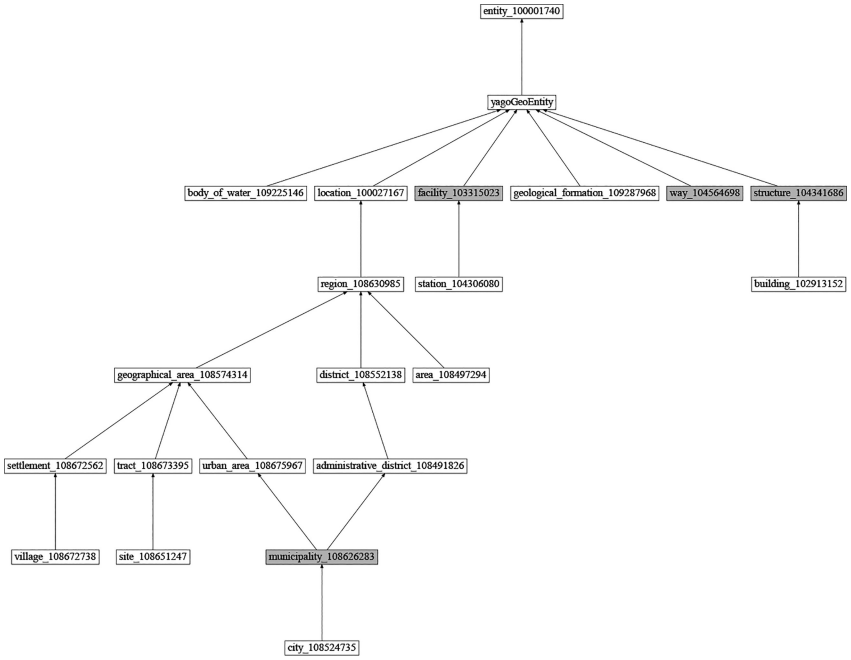


Figure A1 `yagoGeoEntity` branch of the type hierarchy.

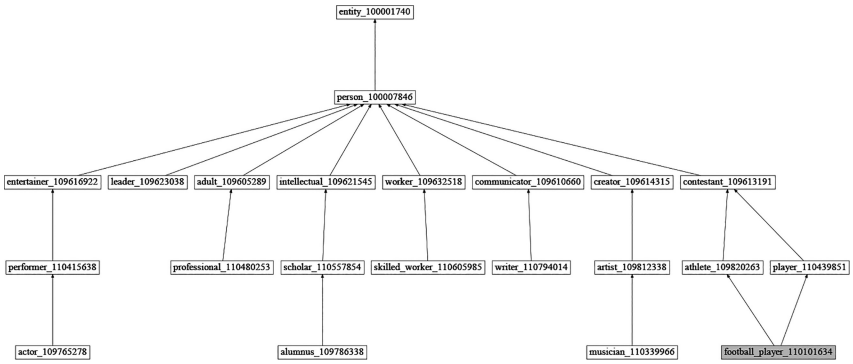


Figure A2 `person` branch of the type hierarchy.

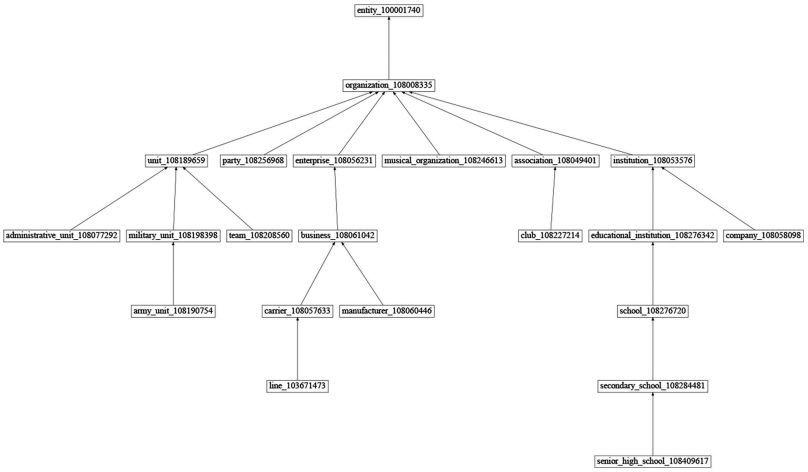


Figure A3 `organization` branch of the type hierarchy.

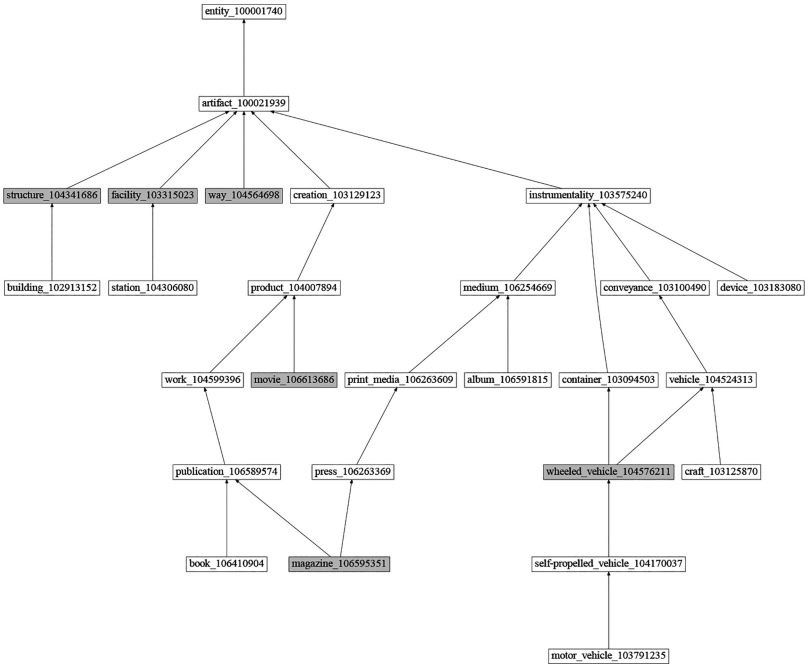


Figure A4 `artifact` branch of the type hierarchy.

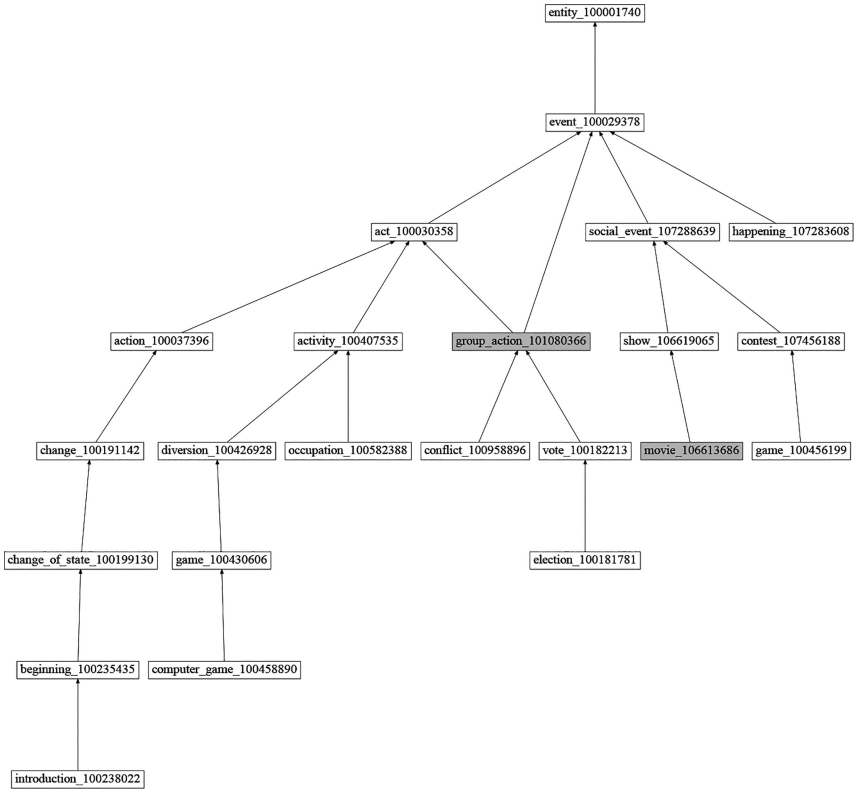


Figure A5 `event` branch of the type hierarchy.

A2 Type-specific Performance Visualization

The following plots depict the performance of each of the aforementioned approaches with respect to individual types (cf. Figure A6–A8). Here, we color a node green if the concerned approach scores more than 0.6 in macro averaged F1 (computed over 10 examples in the test set) for the represented type.

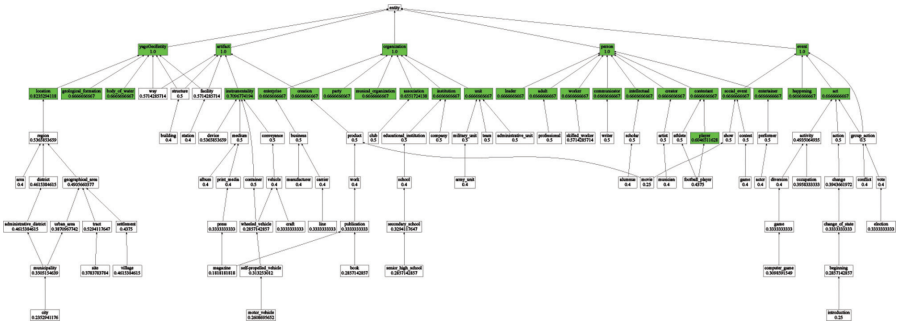


Figure A6 $NaiveBayes$ type-specific performance.

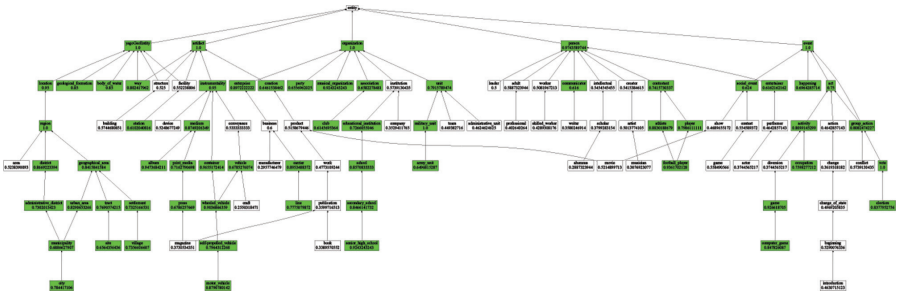


Figure A7 $Elberrichi$ type-specific performance.

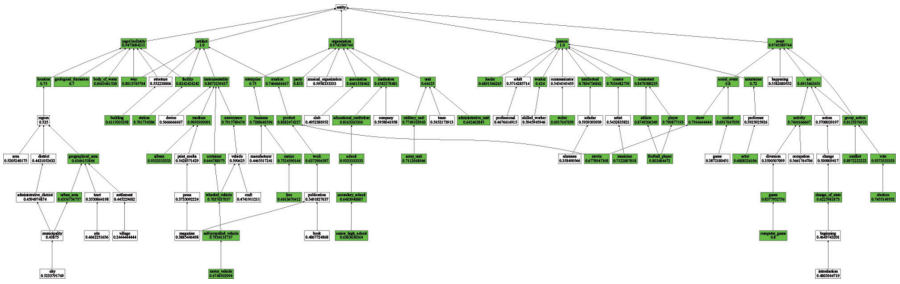


Figure A8 $SemanticFingerprint$ type-specific performance.

References

[1] C. Alec, C. Reynaud-Delaitre, and B. Safar. An Ontology-Driven Approach for Semantic Annotation of Documents with Specific Concepts. In *The Semantic Web. Latest Advances and New Domains. 13th ESWC 2016*, pp. 609–624, Heraklion, Greece, May 2016. Springer.

[2] M. Allahyari, K. J. Kochut, and M. Janik. Ontology-based text classification into dynamically defined topics. In *2014 IEEE International Conference on Semantic Computing*, pp. 273–278, June 2014.

[3] T. Berners-Lee, W. Hall, J. A. Hendler, K. O'Hara, N. Shadbolt, D. J. Weitzner, et al. A framework for web science. *Foundations and Trends® in Web Science*, 1(1):1–130, 2006.

[4] Z. Elberrichi, A. Rahmoun, and M. A. Bentaallah. Using WordNet for Text Categorization. *Int. Arab J. Inf. Technol.*, 5:16–24, 2008.

[5] J. R. Firth. A synopsis of linguistic theory, 1930–1955. 1952–59: 1–32, 1957.

[6] M. Fleischman and E. Hovy. Fine grained classification of named entities. In *Proceedings of the 19th International Conference on Computational Linguistics – Volume 1*, COLING '02, pp. 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[7] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[8] J. Hoffart, D. Milchevski, and G. Weikum. Stics: Searching with strings, things, and cats. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR'14, pp. 1247–1248, New York, USA, 2014. ACM.

[9] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pp. 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[10] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pp. 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[11] R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. *CoRR*, abs/1412. 1058, 2014.

[12] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv: 1607.01759*, 2016.

[13] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[14] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pp. 2267–2273, 2015.

[15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[16] J. Lilleberg, Y. Zhu, and Y. Zhang. Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pp. 136–140, July 2015.

[17] X. Ling and D. S. Weld. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pp. 94–100. AAAI Press, 2012.

[18] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems – Volume 2*, NIPS'13, pp. 3111–3119, USA, 2013. Curran Associates Inc.

[20] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[21] A. Rahman and V. Ng. Inducing fine-grained semantic classes via hierarchical and collective classification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 931–939, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

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

[23] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A core of semantic knowledge–unifying WordNet and Wikipedia. In *16th International World Wide Web Conference (WWW 2007)*, pp. 697–706. ACM, 2007.

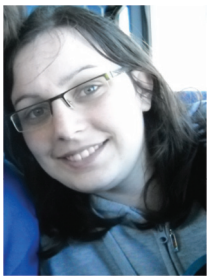
[24] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[25] M. A. Yosef, S. Bauer, J. Hoffart, M. Spaniol, and G. Weikum. Hyena: Hierarchical type classification for entity names. In *Proceedings of COLING 2012: Posters*, pp. 1361–1370. The COLING 2012 Organizing Committee, 2012.

[26] M. A. Yosef, S. Bauer, J. Hoffart, M. Spaniol, and G. Weikum. HYENA-live: Fine-Grained Online Entity Type Classification from Natural-language Text. In *5st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Proceedings of the Conference System Demonstrations, 4–9 August 2013, Sofia, Bulgaria*, pp. 133–138. The Association for Computer Linguistics, 2013.

Biographies

Govind is a post doctoral researcher at University of Caen Normandy, France. His research interests include entity-level analytics, event impact analytics and Web Science. He works on analyzing various aspects of societal events on the Web such as event diffusion prediction into the foreign language communities.



Céline Alec is an assistant professor at University of Caen Normandy, France. She is a member of the HULTECH team in the GREYC lab. Her research interests are in the area of Web science, semantic Web, ontologies, Linked Open Data, knowledge engineering and knowledge acquisition.

Marc Spaniol is a full professor at University of Caen Normandy, France. He is co-organizer of the Temporal Web Analytics Workshop (TempWeb) series. His research interests are in the area in the field of Web science, Web data quality, temporal Web analytics and knowledge evolution.