Undefined 1 (2009) 1–5
IOS Press

Measuring Quality of Evolution in Diachronic Web Vocabularies Using Inferred Optimal Change Models

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country

Albert Meroño-Peñuela [a,b], Christophe Guéret [a,b], and Stefan Schlobach [a]
[a] *Department of Computer Science, VU University Amsterdam, De Boelelaan 1081a, 1081HV Amsterdam, NL*
E-mail: {albert.merono, c.d.m.gueret, k.s.schlobach}@vu.nl
[b] *Data Archiving and Networked Services, Anna van Saksenlaan 10, 2593HT Den Haag, NL*
E-mail: {albert.merono, christophe.gueret}@dans.knaw.nl

Abstract. The Semantic Web uses various commonly agreed vocabularies to enable data from various sources to be effectively integrated and exchanged among applications. In this design, a critical point is the arbitrariness in which these vocabularies can change in subsequent versions. New vocabulary versions reflect changes in the domain, meet new user requirements, and address pitfalls. However, these new versions have an impact in the workflow of publishers of Linked Open Data (LOD), who need to sync their datasets with the new vocabulary releases to avoid ramifications. Predictability of changes in diachronic Web vocabularies is thus highly desired. How predictable are these vocabulary changes in practice? In a longer term, how can we measure the quality of evolving Web vocabularies, and discern between those that "evolve conveniently", and those that change on an arbitrary, even harmful, basis? In this paper, we propose a metric to automatically measure the quality of the evolution of Web vocabularies, based on the performance of inferred optimal change models from past vocabulary versions using well understood evolution predictors. We apply this metric to 139 vocabulary chains from various Semantic Web sources, finding that 39.80% of them evolve in a highly predictable manner.

1. Introduction

Vocabularies play a central role in the workflow of LOD dataset publishers, allowing them to semantically describe and link their data. These vocabularies are revised and regularly republished in new versions in order to "reflect changes in the real world, changes in the user's requirements, and drawbacks in the initial design" [16]. For example, `schema.org` has released 23 different vocabulary versions between 2012 and 2015[1] that have added a number of features[2]. These new vocabulary versions leave LOD datasets using them out of sync, and data publishers need to manually revise new vocabulary versions regularly, and conveniently adapt their datasets.

The impact of new vocabulary versions on the dataset maintenance work of LOD publishers is difficult to assess. Different types of changes affect dataset maintenance differently. For example, vocabularies updated once a year that only change the literals of their `rdfs:comment` relations likely cause less dataset syncing work than the monthly release of the Gene Ontology[3], whose complex changes are difficult to track and cascade to client datasets [15]. In such cases, out-of-sync client datasets create ramifications with old vocabulary versions and miss new features, keep bugs, and generally underperform with respect to the up-to-date vocabulary version. Moreover, the more regular and predictable these vocabulary changes are, the easier LOD publishers will restore sync of their datasets with them [17]. Predictability of changes is, thus, a

[1] See `http://lov.okfn.org/`
[2] See `http://schema.org/docs/releases.html`
[3] See `ftp://ftp.geneontology.org/go/ontology-archive/`

desired characteristic of vocabularies that change over time.

The observation of Web data of dubious quality has given raise to various methods of *Web data quality* assessment. Data quality is commonly conceived as "fitness for use by data consumers" [19], and metrics for measuring quality of Semantic Web data in various dimensions are being deployed [21]. However, none of these metrics are concerned with diachronic[4] vocabularies. The quality of diachronic vocabularies is difficult to estimate, and only manually assessed at best. Currently, no Web data quality metric quantifies the predictability of changes in a *vocabulary version chain*, thus leaving the quality of their evolution and their impact on client datasets undetermined.

So, what is the quality of the evolution processes of Web vocabularies? Are changes introduced in a revision sensible? Are current Web vocabularies evolving in a predictable and coherent way? How can we approach the measurement of such quality evolution? To answer these questions, in this paper we introduce a metric on the *quality of the evolution processes of diachronic Web vocabularies*. To do so, we first find optimal models of change from past versions in a vocabulary chain, using state-of-the-art machine learning tools [15] and well understood ontology evolution predictors [17]. We consequently use the performance of these change models as a quality metric for diachronic vocabulary evolution. Our basic assumption is that a good quality evolution is an evolution that can be learned from data, and that is coherent with our current understanding of ontology evolution. Alternative evolution processes might not be harmful, but can lead to radical and unpredictable changes, making maintainability harder. The contributions of the paper are:

- We generalize an existing change learning method for biomedical ontologies into a domain agnostic method applicable to any Linked Data vocabulary.
- We define a vocabulary evolution quality metric based on the performance of inferred optimal change models.
- We apply this metric to study the quality of Web evolving vocabularies for 669 versions organized in 139 vocabulary chains retrieved from various Web sources. We find that 39.80% of the evaluated vocabulary chains score above 0.9, 36.10% do so between 0.5 and 0.9, and 25.10% display random predictability.
- We discuss the vocabulary characteristics of diachronic Web vocabularies that make them score high in this metric, and the advantages and disadvantages of assessing evolution quality with this method.

[4] Diachronic means developing and evolving over time.

2. Related Work

The Semantic Web has given recent attention to ways for measuring quality of its data [2]. In their recent survey, [21] show that quality of Semantic Web data has been given various definitions and metrics. These metrics are: (i) often defined in a perspective-neutral way, as the *degree to which data fulfills quality requirements* [6]; and (ii) applied to specific definitions of *datasets*. To the best of our knowledge, no previous research has established vocabulary evolution requirements (and thus proposed no related metrics) or considered diachronic datasets.

Changes in Web vocabularies have been investigated by studying their changing semantics, mostly by comparing two subsequent concept or vocabulary versions. [5] propose a method based on clustering similar instances to detect changes in concepts. [7] use Description Logics to calculate differences between ontologies (so-called *semantic diffs*). [20] define the semantics of concept change and drift, and how to identify them. However, these methods (i) infer the *type* of occurring changes only to a limited extent; and (ii) do not consider the *full length* of evolving vocabulary chains. Approaches tackling the latter exist in *ontology evolution*, which studies “the timely adaptation of an ontology and consistent propagation of changes to dependent artifacts” [1]. An important result of ontology evolution is that the starting need for modifying a Web vocabulary can be captured by *structure-driven*, *data-driven* and *usage-driven* features [17]. [15] propose a successful method to model and predict enrichment of classes in biomedical ontologies, by using supervised learning on past ontology versions , using [17] features to design good predictors of change. The need of modelling change in dynamic Web vocabularies in application areas of the Semantic Web has been stressed, particularly in the Digital Humanities [13] and Linked Statistical Data, where concept comparability over time [3,14] is key. Dividino et al. [4] have found that only 35% of LOD schema data remains stable over a year in significant samples, proving that LOD publishers do update their datasets frequently.

3. Change Models for Diachronic Web Vocabularies

Our core idea is that the performance of optimal change models inferred from diachronic Web vocabularies can be used as an indicator of the quality of their evolution. As shown by [15], knowledge encoded in past versions of Web vocabularies can be used to build performant models of change. These models can be used, for instance, to predict which parts of a vocabulary will suffer changes in a forthcoming version. Such change models work because predictors that influence the evolution of Web vocabularies are well understood [17]. Intuitively, a change model that is learned with high performance will characterize a high quality evolving process that can be explained with the chosen evolution predictors. Conversely, a poorly performant model will be related to a rather arbitrary evolution process.

In this section we describe a method that infers optimal change models from arbitrary Web vocabulary chains represented as Linked Data, using supervised learning. To do so:

- *(a)* we use an existing change heuristic [20] to measure pairwise concept change;
- *(b)* we detail the specific features used, based on [17] change capturing predictors;
- *(c)* we generalize the idea of supervised learning of change models in diachronic Web vocabularies from [15], by extending it to any type of change, vocabulary and domain, and by selecting the optimal (most performant) model learned; and
- *(d)* we build a *quality of evolution metric for diachronic Web vocabularies* based on the performance of such model.

3.1. Change Heuristic

We base our definition of change in Web vocabularies on the framework proposed by [20]. They define the *meaning of a concept* C as a triple (label(C),int(C),ext(C)), where label(C) is a string, int(C) a set of properties (the *intension* of C), and ext(C) a subset of the universe (the *extension* of C). To address concept identity over time, they assume that the intension of a concept C is the disjoint union of a rigid and a non-rigid set of properties (i.e. $(int_r(C) \cup int_{nr}(C))$). Then, a concept is uniquely identified by some essential properties that do not change. The notion of identity allows the comparison of two variants of a concept at different points in time, even if a change on its meaning occurs. If two variants of a concept at two different times have the same meaning, there is no concept change. We define intensional, extensional, and label similarity functions $sim_{int}, sim_{ext}, sim_{label}$ in order to quantify meaning similarity. These functions have range $[0, 1]$, and a similarity value of 1 indicates equality. A concept has then *extensionally changed* in two of its variants C' and C", if and only if, $sim_{ext}(C', C'') \neq 1$. Intensional and label change are defined similarly.

3.2. Feature Set

Features affecting the evolution of diachronic Web vocabularies can be **structure-driven**, **data-driven**, and **usage-driven** [17], and we define them conforming with these criteria. *Structure-driven* features are derived from the structure of the ontology (e.g. if a class has a single subclass, both should be merged); and measure the location and the surrounding context of a concept in the dataset schema, such as children concepts, sibling concepts, height of a concept (i.e. distance to the leaves), etc. We define these properties with a $maxDepth$ threshold to avoid cycles (e.g. direct children, children at depth one, two, etc.). A concept is considered to be a child of another if they are connected by a user-specified property (e.g. `rdfs:subClassOf`). We use *direct children* (descendants at distance 1) [*dirChildren*], *children at depth* $\leq maxDepth$ [*dirChildrenD*], *direct parents* (concepts this concept descends from) [*parents*], and *siblings* (concepts that share parents with this concept). *Data-driven* features are derived from the instances that belong to the ontology (e.g. if a class has many instances, the class should be split); and measure to what extent a concept in the schema is used in the data. A data item in a Linked Dataset is considered to be using a concept of the schema if there is a user-defined membership property linking the data item with the concept (e.g. `dc:subject` or `rdf:type`). We use *members of this concept* [*dirArticles*] and *total members considering all children at depth* $\leq maxDepth$ [*dirArticlesChildrenD*] as membership features. *Usage-driven* features are derived from the usage patterns of the ontology in the system it feeds (e.g. remove a class that has not been accessed in a long time). Finally, we define a set of hybrid features that combine some of the previous ones (e.g. ratio of members per number of direct children) [*ratioArticlesChildren*].

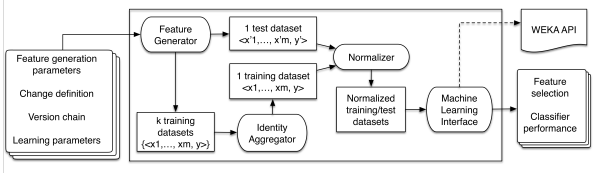


Fig. 1. Optimal change model learning pipeline. Arrows show the data flow through the modules.

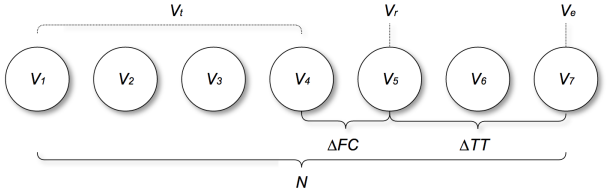


Fig. 2. Training and test datasets for $N = 7$, $\Delta FC = 1$ and $\Delta TT = 2$ in a version chain O.

As described in the next section, these features are generated for all concepts and schema versions, but only used depending on their ranking in the feature selection process.

3.3. Pipeline

Figure 1 shows the change model learning pipeline. Taking input $\{$*Feature generation parameters, change definition, version chain, learning parameters*$\}$, the system returns output $\{$*Feature selection, classifier performance*$\}$.

First, the **Feature Generator (FG)** generates k training datasets and one test dataset, using: *(a)* a *version chain* O containing N versions of a Web vocabulary; *(b)* the feature generation parameters **ΔFC** (which sets the version training concepts will be compared to) and **ΔTT** (which sets the version testing concepts will be compared to); and (c) a *change heuristic*. Then, k training datasets and the test dataset are built by the FG as shown in Figure 2. The parameters N, ΔFC and ΔTT are used to determine which versions will play the role of $\{V_t\}$, V_r and V_e. $\{V_t\}$ is the set of *training versions*, which are used to build the training dataset. V_r is the *reference version*, against which all versions in $\{V_t\}$ are compared, using the change heuristic provided as input. V_e is the *evaluation version* and is used to build the test dataset analogously. V_e is set by default to the most recent version. In order to preserve identity of learning instances, the **Identity Aggregator (IA)** matches concepts in the k training datasets and merges their features into one individual, modifying the dataset dimensionality accordingly. The training and test datasets are then ingested by the **Normalizer (Norm)**, which adjusts value ranges, recodes feature names and types, and discards outliers; followed by the **Machine Learning Interface (MLI)** for the feature selection and classification tasks. These are done in a generic way, using the state-of-the-art machine learning algorithms of the WEKA API [9]. Finally, the *learning parameters* are used here to learn change models using: *(a)* a feature selection algorithm; *(b)* a relevance threshold t to filter the selected features; and *(c)* the list of classifiers to be trained. The MLI runs the chosen feature selection algorithm and trains the chosen subset of WEKA classifiers (all by default), evaluating models and storing results.

3.4. Quality of Evolution Metric

For all classifiers C for a given Web vocabulary version chain O at the output of the MLI, we select the optimal classifier $C_k \in C$ with the maximal area under the ROC curve (a standard classifier performance metric), $roc(C_k) > roc(C_i) \forall C_i \in C$. The *quality of evolution metric* QoE is then just the $roc(C_k)$, $QoE(O) = roc(C_k)$. ***QoE*** is thus simply defined as the performance of the optimal inferred change model for a version chain O. Intuitively, the more performant an optimal change model is for a version chain O, the better O's evolution can be explained by the evolution predictors [17]. We take this as a proxy for the quality of evolution. We will discuss implications of this definition in the discussion section. The assumption behind this choice is that evolution processes that can be learned from data using well-understood evolution features are more desirable than evolution processes that rely on radical changes that cannot be explained by any known feature set. While these alternative evolution processes might not be harmful, their displayed behaviour of radical changes and major refactorings make them stand out from our current explanations of ontology evolution, which is based on regular and predictable changes. In fact, QoE is, by design, a perfect metric to distinguish between these two kinds of evolution processes, and we claim that high QoE scores characterise good evolutions as learnable evolutions, which is a desirable feature for usefulness.

4. Measuring Quality of Evolution

We apply our method to calculate the quality of evolution metric *QoE* for 139 Web vocabulary chains retrieved from the Web, totalling 669 vocabulary versions. We describe the collection and nature of the input data, the experiment setup, and the evaluation criteria, and show the results. We evaluate: *(a)* the performance of the optimal change models learned on training and unseen data for all 139 chains; and *(b)* characteristics of the version chains that score higher on *QoE*. Validating the *QoE* metric is difficult due to the lack of evolution quality benchmarks. There are, however, two reasons indicating its validity: *(1)* it is based on well understood change features [17], notions of change [20] and learning methods [15]; and *(2)* by application we show that it provides excellent results.

4.1. Input Data

We use a set of 139 multi- and interdisciplinary Web vocabulary version chains represented as Linked Data. We classify these 139 version chains in four groups: (1) a version chain of the DBpedia ontology with its latest 8 versions (**DBpedia**); (2) a version chain of the Dutch historical censuses dataset, with its latest 8 versions (**CEDAR**)[5]; (3) 3 reconstructed version chains with ontologies retrieved from 637 public SPARQL endpoints in the Linked Open Data cloud, with at least 3 versions each (**SPARQL**); and (4) 134 version chains from Linked Open Vocabularies[6] with at least 3 versions each (**LOV**). Each version within these chains consists of RDF triples with schema, instances, and labels.

The version chain of the DBpedia ontology [12] is a community-curated formalization of all classes and properties describing DBpedia content. Instances are resources of DBpedia which have some class of the ontology as `rdf:type`. The set of labels are the `rdfs:label` literals attached to the classes of each versioned ontology. In the version chain of the Dutch historical censuses dataset (CEDAR), the classification is a SKOS hierarchy of HISCO occupations reported in each version. Instances are census observations of people having one of these HISCO occupations as `cedar:occupation`. The set of labels are the `skos:prefLabel` (Dutch) literals used in the census to describe these occupations in each specific version. The version chains containing ontologies retrieved from the Linked Data cloud (SPARQL) are retrieved by querying the 637 public SPARQL endpoints listed in `http://datahub.io/`. This returns 49,379 ontologies with at least one previous version (`owl:priorVersion`), and we use this property to reconstruct their version chains. We discard all non-dereferenceable and non-parseable version URIs, and we prune all chains with less than 3 versions, resulting in 3 ontology chains (`geonames`, `fao` and `lingvoj`). Finally, we obtain 134 version chains from Linked Open Vocabularies (LOV), a repository of Semantic Web vocabularies.

4.2. Experimental Setup

Our evaluation process is two-fold. First, we assess the quality of our features as concept change predictors, and we choose the most performing ones. We do this via *feature selection*. Second, we use these selected features for learning, and we evaluate quality of the resulting classifiers on predicting concept change. To evaluate classifiers we follow a simple approach: we compare the predictions made by the classifiers with the actual concept change going on in a next dataset version. To do this, we use the test dataset V_e produced after setting the parameter ΔTT. Since we compare predictions with unseen labeled data, we know whether the predictions are correct or not.

Since more versions are available in the version chains of CEDAR and DBpedia, we execute several learning tasks adding more past versions to $\{V_t\}$ incrementally. We study how this impacts prediction of change in V_i. We also run a learning task considering all versions, and we use the trained classifiers to predict change in the most current version.

For assessing model performance we use the standard performance measures of precision, recall, f-measure, and area under the ROC curve. We perform a two-fold evaluation. On one hand, we evaluate the quality of the models produced without making any predictions and using 10-fold cross-validation with the training data. On the other hand, we use the same indicators to evaluate the classifiers' prediction performance using the unseen test datasets V_e/V_i. We compare our results to a random prediction baseline.

4.3. Results

siblings, *dirArticlesChildrenD2* and *ratioArticlesChildren*; and *dirChildren*, *silbings* and *dirChildrenD2* are the top-3 selected feature sets for CEDAR and DB-

[5] See `http://cedar-project.nl/`

[6] See `http://lov.okfn.org/dataset/lov/`

	fao	geonames	lingvoj	LOV (avg.)
Precision	.751	.438	.95	-
Recall	.765	.662	.947	-
F-measure	.744	.527	.937	.922
ROC area	.844	.5	.792	.566

Table 1

10-fold CV scores in the version chains from LOD SPARQL endpoints and Linked Open Vocabularies.

pedia, respectively, by the `Relief` algorithm [10], included in the WEKA API[7]. We observe that data-driven features are systematically selected in the CEDAR data instead of structure-driven properties. Conversely, we observe a clear preference for structure-driven properties in the DBpedia data. We execute our approach six times in the Dutch historical censuses and the DBpedia version chains, adding one Linked Dataset version to $\{V_i\}$ and shifting V_i forward once each time. We identify each experiment with the year/-timestamp of the version to be refined. Figure 6 shows the results.

Selected features for the 3 version chains retrieved from the SPARQL endpoints and the 134 version chains of LOV are available at `<supplemental-material>`. Predictive models for these datasets are learned with different results, as shown in Table 1. The quality of the prediction using learned models for the SPARQL vocabularies is very high in the `fao` and `lingvoj` version chains, but almost as bad as random in `geonames`. Explanation for such results are detailed in the next section. Results for the LOV version chains can be found in detail at `<supplemental-material>`.

Figure 3 shows a histogram of all 139 optimal change models, one per version chain, and their performance frequency, given by their area under the ROC curve (i.e. their scores on the *QoE* metric). We observe that performant models, i.e. with $QoE > 0.9$, can be built for 39.80% of the evaluated vocabulary chains. Conversely, for 25.10% of the datasets no good model could be built, being random at best. For the remaining 36.10% of the datasets only modest models could be found ($0.5 < QoE < 0.9$).

4.4. Characterization of Quality Version Chains

In this section we study what specific characteristics of the input version chains have a relationship with the performance of the learned models. To investigate this, we compute, for each version chain, a set of *version chain characteristics* that include: size of the chain (*totalSize*) in number of triples; number of versions in the chain (*nSnapshots*); average time gap (in days) between the release date of each version (*avgGap*); average size of each version (*avgSize*); number of inserted new statements between versions (*nInserts*)[8]; number of deletes (*nDeletes*); number of common statements (*nComm*); is the schema a tree or a graph (*isTree*); maximum tree depth among versions (*maxTreeDepth*); average tree depth (*avgTreeDepth*); number of instances (*totalInstances*); ratio of instances over all statements (*ratioInstances*); number of structural relationships (*totalStructural*); and ratio of structural relationships over all statements (*ratioStructural*). First, we use regression to analyse which of these predict better the performance of the optimal change model, using the *QoE* score as a response variable, shown in Figure 4[9]. We find that, under the null hypothesis of normality and non-dependence, the predictors *nSnapshots*, *avgTreeDepth*, *ratioStructural*, *ratioInserts* and *ratioComm* (discarding *ratioDeletes* and *ratioComm* due to multi-colinearity) are good explanatory variables with respect to *QoE*. Secondly, we use multinomial logistic regression to find predictors of the optimal classifier type. A simulation is shown in Figure 5[7]. We find that *avgGap* is influential at selecting a tree classifier instead of Bayes, and *totalSize* is influential at selecting functions and rules-based classifiers instead of Bayes. Figure 5 shows a simulation on how these predictors[7] influence the choice of the different classifier families. All classifier families will be less likely chosen when the release time gap decreases, ex-

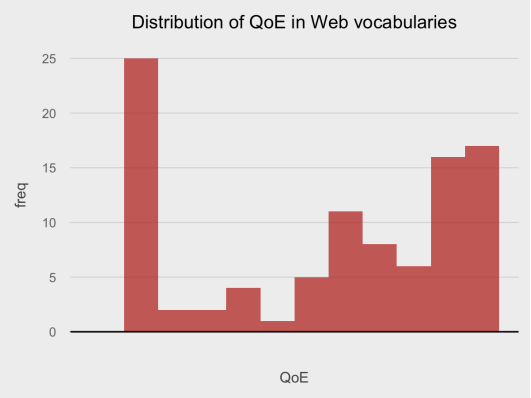


Fig. 3. Distribution of performance of learned optimal change models.

[7] Detailed feature selection at `<supplemental-material>`

[8] Insertions and deletions measured with UNIX's `diff`.

[9] Additional details at `<supplemental-material>`.

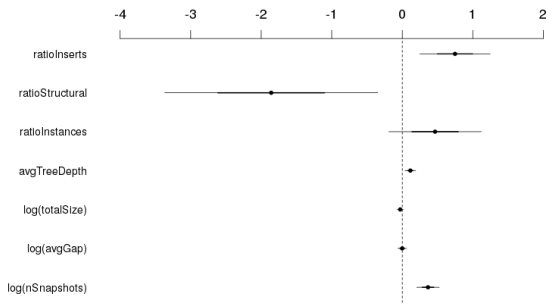


Fig. 4. Coefficients and errors of the best linear regression model built to find predictors of quality evolving vocabularies.

cept for tree-based classifiers; meaning that more frequent releases will favour most models. Interestingly, ratios on instance and schema data will influence the best classifier type in an inverse way: more instance data will favour tree-based and rules classifiers, while more schema data will favour bayesian classifiers. We discuss these results in the next section.

5. Discussion

In this section we discuss our findings, by *(a)* observing specific correctly predicted changing concepts; *(b)* arguing the different classifier performances; *(c)* investigating the meaning of schema characteristics that are related with high *QoE* scores; and *(d)* understanding the meaning of our proposed quality of evolution metric *QoE*.

`http://cedar.example.org/ns#hisco-06` is an example concept from the CEDAR schema versions predicted to change which in fact did: the class of "medical, dental, veterinary and related workers". Its features are stable except those data-driven: the class abruptly declines from 841 instances to 68, 143, 662 and 110; while structure-driven properties like number of children (4) or siblings (9) remain stable. Similarly, the DBpedia concept `http://dbpedia.org/ontology/CollegeCoach` is correctly predicted to change, having a linear increase of pointed Wikipedia articles (2787, 3520, 4036, 4870...); however, its siblings remain stable (21, 21, 23, 23) until it gets a new parent and its siblings suddenly explode (23, 344). Therefore, it is easy to see why data- and structure-driven features are related to predictable schema evolutions.

Although Logistic, MultilayerPerceptron and tree-based classifiers have good eventual performances, NaiveBayes classifier shows consistent results in all change prediction experiments. Similar behavior and results have been described [15]. Interestingly, we observe how the non-overfitting tendency of NaiveBayes is an advantage if the classifier is trained with more past versions (*nSnapshots*): MultilayerPerceptron, for instance, predicts better with less data (f-measures from 0.82 to 0.30), but with more versions NaiveBayes wins (0.72 to 0.84). However, performance is poorer in some versions (e.g. 1889 and 1930 of CEDAR, 2010 and 2011 of DBpedia). Historical research shows that those versions suffered major revisions almost from scratch [11], which would make their changes harder to predict. Still, the metric proves to be useful on detecting these coherence data-issues. Figure 6 shows that these classifiers outperform the random baseline.

Predictors described in the previous section on explaining Web vocabulary version chains that score high in the *QoE* metric (see right-hand side vocabularies of Figure 3) lead to three important observations: (1) a longer version history in a vocabulary makes its evolution more predictable; (2) schema information is more important than instance information for change modelling; and (3) inserting new statements and leaving the existing ones in a new release helps more in preserving change consistency than removing old statements. In addition, the behaviour of predictor *avgGap* (see Figure 5) suggests that a vocabulary will score higher in *QoE* if the time between version releases is short. Intuitively, vocabularies meeting these criteria will have higher chances of having performant optimal change models related to [17]'s features, and thus score higher in the *QoE* metric. This has a logical sense: more frequent and numerous version releases increase the amount of past knowledge to learn from; while the addition (and scarce removal) of structural statements matches the notion that evolution is better predictable if done smoothly and incrementally.

Remarkably, 57.73% of all 139 evaluated chains score $QoE > 0.8$ as shown in Figure 3, and thus more than half of the vocabulary chains display a highly predictable change and a smooth evolution process. PROV [8] is an example of such a vocabulary: with 8 releases equally spread since 2012, its URIs, names and features evolve incrementally (e.g. new structural statements for `prov:wasInvalidatedBy` and `prov:revisedEntity`), while the core conceptual model (`prov:Entity`, `prov:Agent` and `prov:Activity` and their relations) remains stable. Introduction of new features is smooth, and refactoring or removal of statements from previous versions rarely happens. Contrar-

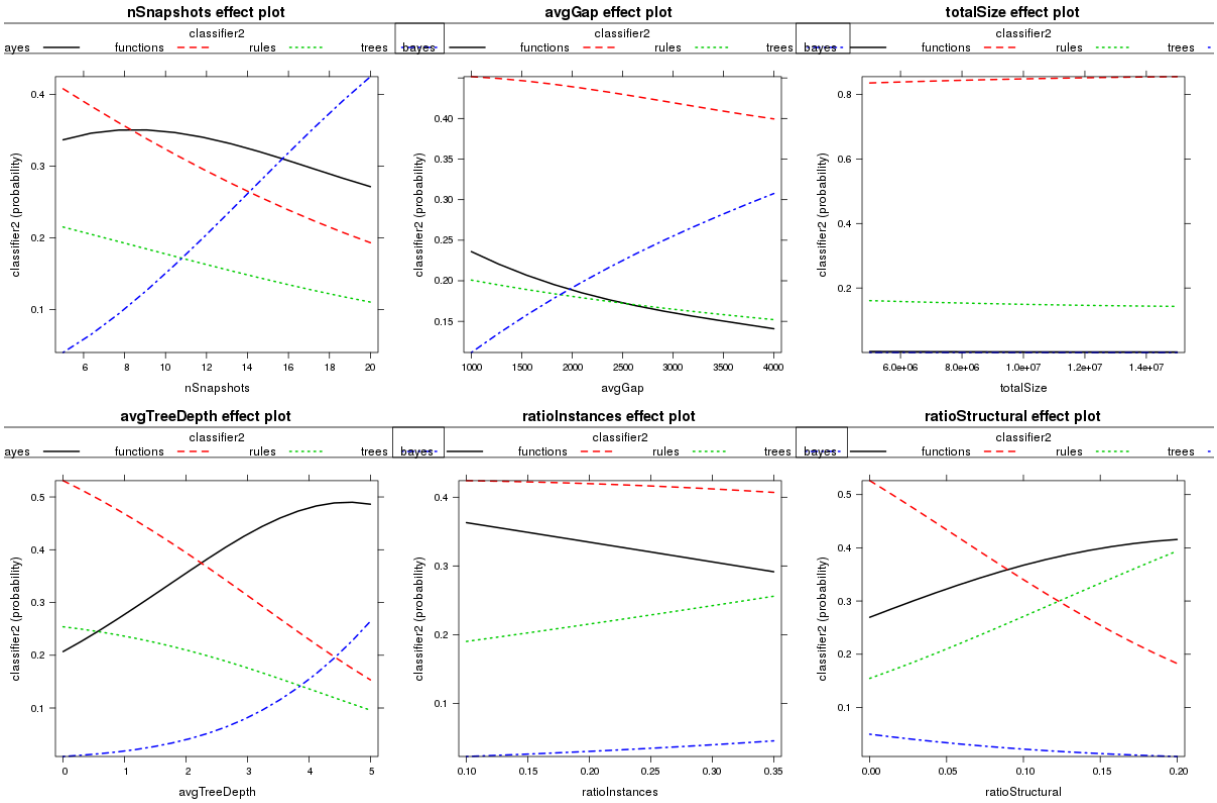
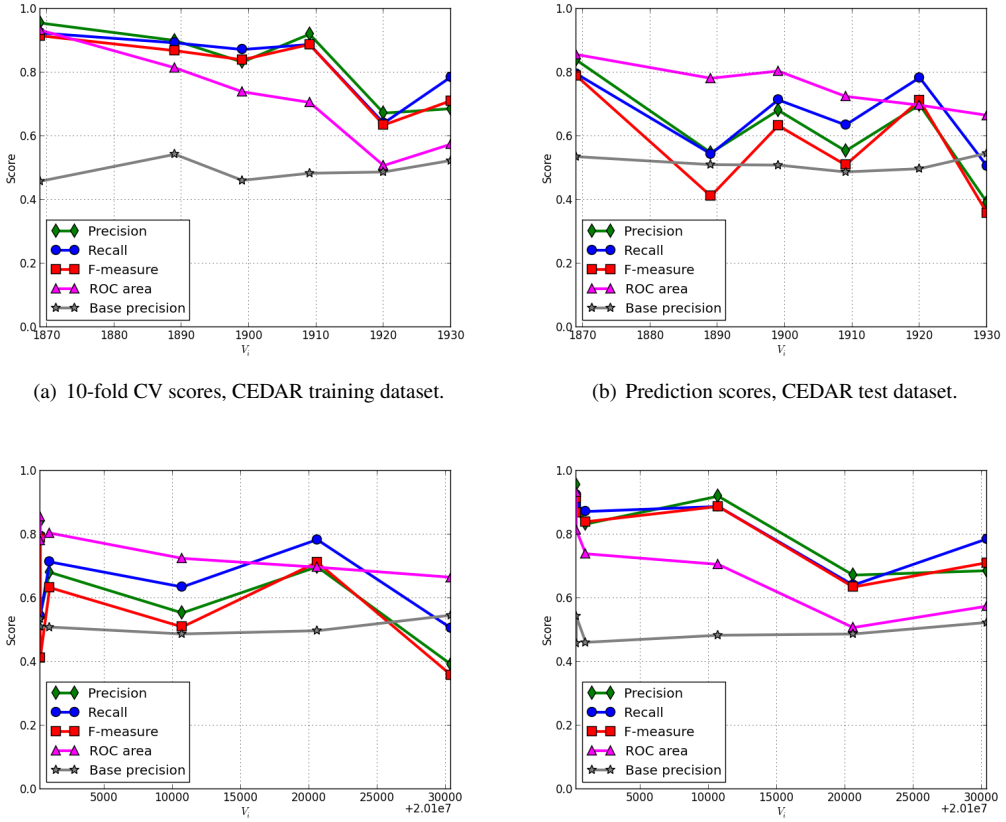


Fig. 5. Simulation of how predictors influence the best classifier chosen using multinomial logistic regression. E.g., *avgGap* shows that smaller time gaps between releases favours almost all classifier types, except those tree based.

ily, for 24.74% of the evaluated chains no performant model could be found, and their change predictability is as good as random ($QoE = 0.5$). The Geonames ontology [18] is an example: 6 of its 11 releases happened in 2006/7, with major refactorings from 2010 onward. Some classes, like `gn:Country` and the URI policy of the feature code scheme, were removed after the first releases or majorly refactored. Additionally, the addition of new features like `gn:historicalName` and `gn:officialName` were only introduced in the last release, and thus difficult to be explained by the optimal change model. What does this mean in terms of the definition of *QoE*? Essentially, the closer a vocabulary scores to 0.5 in *QoE*, the less it is evolving in a way [17]'s features can predict. While this might not be necessarily harmful, it shows misfit with our current understanding of ontology evolution, and reveals radical changes and major refactorings; phenomena that deserve being flagged for later check-up. Notice that scores $0.0 < QoE < 0.5$ and $QoE = 0.5$ indicate inverse and random functions, respectively. Thresholds of *QoE* to distinguish highly performant models are in general arbitrary and depend on the task; but plots like Figure 3 should help to detect these in practice.

6. Conclusion and Future Work

In this paper we have motivated the problem of assessing quality of the evolution processes of diachronic Web vocabularies. Semantic Web ontologies, vocabularies and taxonomies are refined to be adapted to a changing world, leaving client datasets out of sync. Changes in new vocabulary versions either respond to well understood evolution behaviour, or the wild arbitrariness of the Web's freedom, with a whole spectrum in between. The predictability of their changes affects the maintenance work of LOD publishers whose datasets depend on these changing vocabularies. How predictable is the Semantic Web on curating its evolving ontologies and vocabularies? We propose a metric for assessing quality of the evolution of Web vocabularies based on optimal change models inferred from Web vocabulary version chains, finding that 39.80% of



(a) 10-fold CV scores, CEDAR training dataset.

(b) Prediction scores, CEDAR test dataset.

(c) 10-fold CV scores, DBpedia training dataset.

(d) Prediction scores, DBpedia test dataset.

Fig. 6. Average classifier performance in the CEDAR and DBpedia refinement experiment with 6 incremental learning runs. Lines show performance measures varying along them.

a varied sample of 139 chains from the Semantic Web displays a highly predictive evolution. On the other hand, 25.10% of these vocabularies score low in the metric and inherently cause more arduous sync work to LOD publishers.

We plan to extend this work in several ways. First, we will update the study of [17] to find new evolution predictors behind low-scoring vocabularies. Second, we will improve the pipeline's performance to learn optimal change models efficiently. Finally, we will extend the sample of Web vocabulary chains to all known vocabularies in the Semantic Web to provide live monitoring of their evolution quality.

References

[1] B. Motik A. Mäedche and L. Stojanovic. Managing multiple and distributed ontologies in the Semantic Web. *VLCB Journal*, 12(4):286–300, 2003.

[2] Riccardo Albertoni, Christophe Guéret, and Antoine Isaac. Data Quality Vocabulary (First Public Working Draft). Technical report, World Wide Web Consortium, 2015.

[3] Sarven Capadisli. Linked Statistical Data Analysis. In *1st International Workshop on Semantic Statistics (SemStats 2013), ISWC*. CEUR, 2013.

[4] Renata Dividino, Ansgar Scherp, Gerd Gröner, and Thomas Gottron. Change-a-LOD: Does the Schema on the Linked Data Cloud Change or Not? In *Proceedings of the Fourth International Workshop on Consuming Linked Data (COLD2013)*, 2013.

[5] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Conceptual Clustering: Concept Formation, Drift and Novelty Detection. In *The Semantic Web: Research and Applications, 5th European Semantic Web Conference. LNCS 5021*, pages 318–332. Springer, 2008.

[6] Christian Fürber and Martin Hepp. Using Semantic Web Resources for Data Quality Management. In Philipp Cimiano and H.Sofia Pinto, editors, *Knowledge Engineering and Management by the Masses*, volume 6317 of *Lecture Notes in Computer Science*, pages 211–225. Springer Berlin Heidelberg,

2010.

[7] Rafael S. Gonçalves, Bijan Parsia, and Uli Sattler. Analysing Multiple Versions of an Ontology : A Study of the NCI Thesaurus. In *24th International Workshop on Description Logics (DL 2011)*, volume 745. CEUR, 2011.

[8] Paul Groth and Luc Moreau. PROV-Overview. An Overview of the PROV Family of Documents. Technical report, W3C, 2013. http://www.w3.org/TR/prov-overview/.

[9] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[10] Kenji Kira and Larry A. Rendell. The Feature Selection Problem: Traditional Methods and a New Algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI'92, pages 129–134. AAAI Press, 1992.

[11] P.S. Lambert, R.L. Zijdeman, I. Maas, K. Prandy, and M.H.D. van Leeuwen. Testing the universality of historical occupational stratification structures across time and space. In *ISA RC 28 Social Stratification and Mobility spring meeting, Nijmegen*, 2006.

[12] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, SÃűren Auer, and Christian Bizer. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web âĂŞ Interoperability, Usability, Applicability*, 2014. http://www.semantic-web-journal.net/system/files/swj558.pdf.

[13] Albert Meroño-Peñuela. Semantic Web for the Humanities. In *The Semantic Web: Semantics and Big Data, 10th European Semantic Web Conference. LNCS 7882*, pages 645–649. Springer, 2013.

[14] Albert Meroño-Peñuela, Christophe Guéret, Rinke Hoekstra, and Stefan Schlobach. Detecting and Reporting Extensional Concept Drift in Statistical Linked Data. In *1st International Workshop on Semantic Statistics (SemStats 2013), ISWC*. CEUR, 2013.

[15] Catia Pesquita and Francisco M. Couto. Predicting the Extension of Biomedical Ontologies. *PLoS Computational Biology*, 8(9):e1002630, 2012. doi:10.1371/journal.pcbi.1002630.

[16] L. Stojanovic and B. Motik. Ontology Evolution within Ontology Editors . In *Evaluation of Ontology-based Tools Workshop, 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, volume 62. CEUR-WS, 2002.

[17] Ljiljiana Stojanovic. *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe, 2004.

[18] Bernard Vatant and Marc Wick. Geonames ontology documentation. Technical report, Unxos GmbH, Switzerland, 2012. http://www.geonames.org/ontology/documentation.html.

[19] Richard Y. Wang and Diane M. Strong. Beyond accuracy: What data quality means to data consumers. *J. Manage. Inf. Syst.*, 12(4):5–33, March 1996.

[20] Shenghui Wang, Stefan Schlobach, and Michel C. A. Klein. What Is Concept Drift and How to Measure It? In *Knowledge Engineering and Management by the Masses - 17th International Conference, EKAW 2010. Proceedings.*, pages 241–256. Lecutre Notes in Computer Science, 6317, Springer, 2010.

[21] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. Quality Assessment for Linked Data: A Survey. *Semantic Web âĂŞ Interoperability, Usability, Applicability*, 2014. http://www.semantic-web-journal.net/system/files/swj773.pdf.