Sequential Action Patterns in Collaborative Ontology-Engineering Projects: A Case-Study in the Biomedical Domain

Simon Walk*
Graz University of Technology
Graz, Austria
simon.walk@tugraz.at

Philipp Singer*
GESIS - Leibniz Institute for
the Social Sciences
Cologne, Germany
philipp.singer@gesis.org

Markus Strohmaier
GESIS & Univ. of Koblenz
Cologne & Koblenz, Germany
markus.strohmaier@gesis.org

ABSTRACT

Within the last few years the importance of collaborative ontology-engineering projects, especially in the biomedical domain, has drastically increased. This recent trend is a direct consequence of the growing complexity of these structured data representations, which no single individual is able to handle anymore. For example, the World Health Organization is currently actively developing the next revision of the International Classification of Diseases (ICD), using an OWL-based core for data representation and Web 2.0 technologies to augment collaboration. This new revision of ICD consists of roughly $50,000$ diseases and causes of death and is used in many countries around the world to encode patient history, to compile health-related statistics and spendings. Hence, it is crucial for practitioners to better understand and steer the underlying processes of how users collaboratively edit an ontology. Particularly, generating predictive models is a pressing issue as these models may be leveraged for generating recommendations in collaborative ontology-engineering projects and to determine the implications of potential actions on the ontology and community. In this paper we approach this task by (i) *exploring* whether regularities and common patterns in user action sequences, derived from change-logs of five different collaborative ontology-engineering projects from the biomedical domain, exist. Based on this information we (ii) *model* the data using Markov chains of varying order, which are then used to (iii) *predict* user actions in the sequences at hand.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Medical information systems; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Web-based interaction*

Keywords

Markov Chain; Sequential Pattern; State Prediction; Collaborative Ontology-Engineering

1. INTRODUCTION

The complexity of structured knowledge representations, especially in the biomedical domain, has dramatically increased over the last decade. This recent trend is the direct result of the increasing requirements for these ontologies to satisfy, due to a growing field of application. For example, the International Classification of Diseases in its 10th revision (ICD-10) is used to encode patient history data and to compile health-related spending and morbidity as well as mortality statistics for international comparison. To increase the utility of ICD, the World Health Organization (WHO) is currently developing the 11th revision of this classification (ICD-11), using the Internet and Web 2.0 technologies as collaboration platform and an OWL-based core for knowledge representation. This change in knowledge representation will allow for additional information to be stored inside ICD-11. For example, diseases will have (among others) explicitly defined related/affected body parts and diagnostic criteria. Compared to ICD-10, the new revision now contains around $50,000$ diseases and causes of death, thus has roughly tripled in size and is to be developed until 2017.

Due to this increase in complexity, ontologies, such as ICD-11, can no longer be developed by single authorities. Instead, WHO decided to open-up the development process of ICD-11, allowing everyone with access to the Internet to contribute and discuss changes made to the ontology. However, this open and collaborative ontology-engineering process poses many, yet unidentified, problems to tackle and anticipate. For instance, tracking and monitoring user actions or the overall progress of the underlying ontology as well as helping users to identify work tasks, which they have the required expertise to contribute to, are two either computationally expensive or very time consuming tasks. In particular, administrators of collaborative ontology-engineering projects are in need of better tools to understand and augment users when contributing to these projects.

Objective. Our main objective is to predict user actions in collaborative ontology-engineering projects; e.g., the property a user is most likely to edit next. We want to achieve this task by first exploring whether regularities and sequential patterns exist, then building upon these observations for modeling the data and finally, evaluating the prediction accuracy of each model.

Approach. Specifically, we will approach this objective as follows in subsequent order:
(i) Exploring action sequences: First, we investigate whether action sequences based on several dimensions (e.g., sequential properties changed by users as illustrated in Figure 1) exhibit regularities or are emerging in random fashion before we mine and study common sequential patterns in our data.

*Both authors contributed equally to this work.

CIKM'14, November 3–7, 2014, Shanghai, China.
Copyright 2014 ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2662049 .

(ii) Modeling action sequences: Next, we establish our model approach using Markov chains of varying order, allowing us to incorporate our insights from the first research approach. We also present model selection techniques that can be used for testing and evaluating the accuracy of these models.
(iii) Predicting user actions: Subsequently, we fit these models to our data and evaluate each model, giving insights into their predictive power. The models may be leveraged for generating recommendations in collaborative ontology-engineering projects and to determine the implications of potential actions on the ontology and community.

We perform our experiments on five datasets stemming from different biomedical projects (ICD-11, The International Classification of Traditional Medicine (ICTM), The National Cancer Institute Thesaurus (NCIt), The Biomedical Resource Ontology (BRO) and The Ontology of Parasite Lifecycle (OPL); for more details see Section 2).

Contributions. To the best of our knowledge, this paper presents the most detailed analysis of sequential user actions in collaborative ontology-engineering projects in the biomedical domain for predicting future actions. We find (significant) evidence that (i) regularities and (long) sequential patterns do exist and (ii) demonstrate their utility for predicting the action that is most likely to occur next in our datasets.

Our insights not only improve our understanding of how users engage in collaborative ontology-engineering projects but can also potentially improve the workflow of collaborators by, e.g., recommending properties to contributors to edit next. By doing so, we may be able to better leverage the expertise of contributors by steering them into the right direction. Apart from that, practitioners may also be able to enhance the quality of specific parts of the ontology by promoting them to the right users. Having predictive models for user actions will also allow collaborative ontology-engineering project administrators to assess potential actions regarding their implications on the underlying ontology and community.

Structure of this article. We introduce our experimental setup in Section 2 before we explore action sequences in Section 3. We introduce our model approach in Section 4 and apply and evaluate these models in Section 5. We discuss (Section 6) our findings and related work (Section 7) next and conclude our work in Section 8.

2. EXPERIMENTAL SETUP

In this section we first briefly introduce our five datasets, stemming from the biomedical domain, before we elaborate on our specific dataset preparation steps.

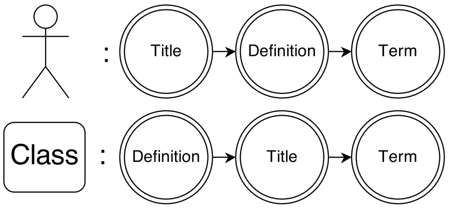


Figure 1: The top row of the figure depicts an exemplary user-based property sequence with properties *Title*, *Definition* and *Term* for a user. This means that the first property that was changed by the user is *Title*, then *Definition* and last *Term*. The bottom row of the figure shows the class-based sequential property path for a class and the same properties *Title*, *Definition* and *Term*. Analogously, the first property that was changed for the class was *Definition*, then *Title* and last *Term*.

2.1 Dataset Description

Table 1 lists the detailed features and observation periods for all datasets used in our analysis. The two largest datasets are ICD-11[1] and the National Cancer Institute Thesaurus (NCIt) [28] with 48,771 and 102,865 classes and 439,299 and 294,471 changes respectively. NCIt is a reference vocabulary for clinical care, translational, basic research and cancer biology. The International Classification of Traditional Medicine (ICTM), which was first intended to be a stand-alone biomedical ontology but was merged with ICD-11 after our observation period, represents a collaborative ontology-engineering project of medium size, with 1,506 classes and a total of 67,522 changes. ICTM is developed by WHO and tries to unify knowledge from traditional medicine practices from China, Japan and Korea. The Biomedical Resource Ontology (BRO) and the Ontology for Parasite Lifecycle (OPL) are two smaller sized collaborative ontology-engineering projects with only 528 and 393 classes and 2,507 and 1,993 changes respectively. BRO is a controlled terminology for describing the source type, areas of research, and activity of biomedical related resources. OPL models the life cycle of a parasite, which is responsible for a number of human diseases.

2.2 Dataset preparation

We extracted sequences from activity logs of the five collaborative ontology-engineering datasets to perform our experiments on. All extracted sequences are either *class- or user-based* (see Figure 1). A class-based sequence depicts a chronology of a specific feature of all changes that were performed *by any user on a single class*. A user-based sequence, analogously, captures the ordered list

[1] `http://www.who.int/classifications/icd/ICDRevision/`

Table 1: Characteristics of the investigated datasets. Note that all datasets differ in size (number of classes and users), activity (number of changes) and observation periods. ICD-11 and ICTM both exhibit changes that were performed automatically and are denoted as *# of bots (changes)* in the table. For our analysis we removed these changes.

		ICD-11	ICTM	NCIt	BRO	OPL
Ontology	# of classes	48,771	1,506	102,865	528	393
	# of changes	439,229	67,522	294,471	2,507	1,993
Users	# of users	109	27	17	5	3
	# of bots (changes)	1 (935)	1 (1)	0 (0)	0 (0)	0 (0)
Duration	first change	18.11.2009	02.02.2011	01.06.2010	12.02.2010	09.06.2011
	last change	29.08.2013	17.7.2013	19.08.2013	06.03.2010	23.09.2011
	observation period (ca.)	4 years	2.5 years	3 years	1 month	3 months

of specific features of changes that were performed *on any class by a single user* for each dataset. Note that we are interested in studying collaborative behavior in this paper and hence, provide an aggregated view on the data based on all users or all classes. Thus, we always work with a set of distinct sequences where each sequence corresponds to one single user (user-based) or one single class (class-based). In a preprocessing step, we pruned all sequences that exhibit less than two elements, for example, if a class was only ever changed by one user, we removed this specific entry from our training set. Note that we have removed all automatic changes performed in ICD-11 and ICTM for our analyses (see Table 1). In Sections 3 and 4, we will closely investigate the following aspects (and thus sequences) of the activity logs:

(i) Users for Classes. These, solely class-based, sequences consist of chronologically ordered lists, where each list captures one class, of users that changed a specific class.

(ii) Change-Types for Classes and Users. Such a sequence contains a chronology of change-types of the performed changes by a specific user on any class (user-based) or the change-types of the performed changes for a specific class by any user (class-based). We aggregated the performed change-types into abstract classes of changes, which was necessary due to the large variety of different change-types present in our investigated datasets. All changes that edit the value of a property of a class have been aggregated (i.e., added property, edited property, deleted property). Analogously, we have aggregated the changes performed on classes (i.e., added class, moved class, removed class, deleted class).

(iii) Properties for Classes and Users. These sequences consist of chronologically ordered lists of properties changed by a specific user of any class (user-based) or the properties changed for a specific class by any user (class-based).

Note that we were not able to conduct the *Change-Types for Classes and Users* and *Properties for Classes and Users* analyses for NCIt. The reason for this is the existence of a specific feature in the ontology-editor that is used to develop NCIt, which allows contributors to queue changes and commit batches of changes simultaneously to the ontology.

3. EXPLORING ACTION SEQUENCES

In this section we explore the nature of our action sequences at hand. We first investigate randomness and regularities in Section 3.1 and then continue to extract common sequential patterns in Section 3.2.

3.1 Randomness and Regularities

To begin with, we are interested in determining whether our data sequences are produced in random fashion or based on some regularities. One common way to investigate randomness in such sequences or time series is to use *autocorrelation* with varying lags [6]. This method builds on Pearson's product-moment correlation coefficient which determines linear relationships between lagged variables. Contrary, in our paper, we work with categorical data in our sequences (e.g., properties) which is why the autocorrelation method is not directly applicable to our problem at hand.

Another way of determining randomness in data sequences is the so-called *runs test* which is also more specifically entitled *Wald-Wolfowitz runs test* [35, 7]. It is a non-parametric test in which the null hypothesis (the sequence was produced randomly; the elements of the sequence are independent to each other) is tested against the alternative hypothesis stating that the sequence was not produced randomly. In particular, the null hypothesis gets rejected if the total number of runs – a run is a series of identical values (e.g., the sequence "AABA" has three runs "AA", "B" and "A") – is too small leading to a clustered arrangement or too large resulting in a systematic arrangement [21]. Predominantly, the test is only suited for sequences with binary or dichotomous observations. O'Brien and Dyck [21] adapted the initial method by proposing a test that is based on a linear combination of the weighted variances of run lengths. This approach can now be extended to also work with categorical observations which is required for our analyses.[2] We exemplarily applied this method on our individual ICD-11 sequences, and can clearly see that a significant proportion of sequences is produced in a non-random way. This is imminent as the null hypotheses regularly gets rejected (p-value below 0.05) – e.g., the null hypotheses gets rejected for more than 60% of all user property sequences. Our observations in this section warrant further investigations of patterns and structural properties in these sequences. Hence, we next focus on investigating how these present regularities in our sequential patterns look like; i.e., we focus on mining common sequential patterns.

3.2 Sequential Pattern Mining

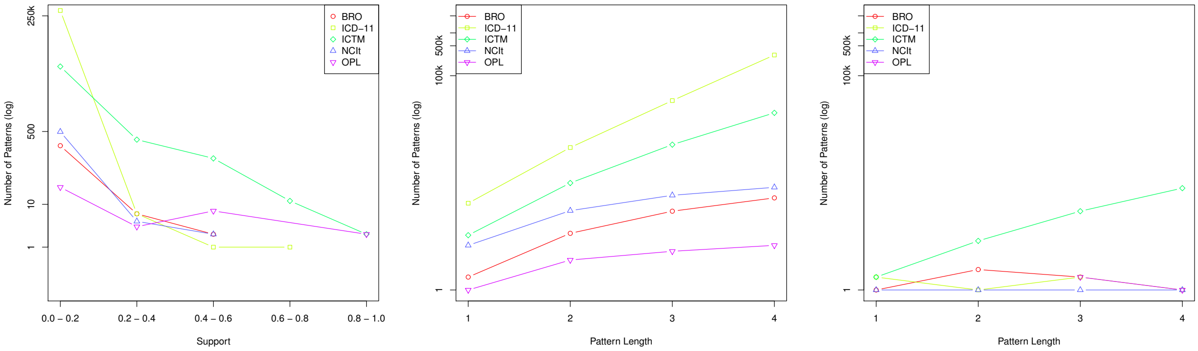
Given our observations made in Section 3.1, we are now interested in actual sequential patterns that account for the regularities in the activity logs. There do exist a variety of algorithms to extract the most frequently used sequential patterns from a set of sequences. We make use of PrefixSpan [22] to investigate commonly used sequential patterns in collaborative ontology-engineering project change-logs, as the algorithm concentrates on expanding (or growing) frequently used patterns and strictly matches only patterns to sequences that are completely identical (i.e., do not exhibit gaps or skipped elements). Support for sequential pattern mining algorithms, a measure to determine how frequent certain patterns are observed in the data, is usually defined as the percentage of all investigated paths that contain a given pattern. Note that all paths have to be chronologically sorted and patterns only consist of succeeding states. For example, the pattern "AB" is *not* present in the sequence "ACBA", as "B" never immediately succeeds "A".

PrefixSpan first scans all available sequences and denotes the number of occurrences for each element in all sequences. It then stores the occurrences and the remainder of the sequences (the suffix) and uses the most frequently used sequential patterns as prefix requirement for the next iteration. Analogously, the prefix is again expanded until a certain level (minimum) support is reached.

We have applied PrefixSpan on the five collaborative ontology-engineering project datasets to see if and to what extent such sequential patterns are present. As can be seen in Figure 2(a), PrefixSpan was able to extract between 5 to 500 patterns for the *Predicting Users for Classes* analysis across all five datasets with a support of 0.2 to 0.4. This means that the identified sequential patterns are present in 20 to 40 percent of all investigated sequences. Figures 2(b) and 2(c) show the number of identified patterns of lengths 1 to 4 for support levels of 0.0 to 0.2 and 0.2 to 0.4. Similar observations could be made for the other analyses.

Given the high number of sequential patterns of lengths 2 to 4 we argue that such patterns play a crucial role in the contributor logs of collaborative ontology-engineering projects at hand. Hence, we believe that there might be some dependence between subsequent

[2] We make an implementation of this method available online at `https://github.com/psinger/RunsTest`. Note though that the method has some limitations. For example, there have to be more than one distinct run length for an element, more than one success run and the number of successes minus the number of success runs of an element has to exceed one. For more details please refer to [21] and the source on github. Hence, we only recommend to perform the test on "somewhat" longer sequences with more runs which is the case for our data at hand.



(a) Patterns per Support

(b) Number of Patterns with length n for Support 0.0 - 0.2

(c) Number of Patterns with length n for Support 0.2 - 0.4

Figure 2: Results of the PrefixSpan analysis on the *Predicting Users for Classes Sequences*: Figure 2(a) shows the number of extracted patterns (y-axis; log-scale) by PrefixSpan for a given support range (x-axis). Support is defined as the percentage of paths that exhibit a certain pattern. For example, the roughly 500 sequential patterns extracted for ICTM with a support level of 0.2 - 0.4 are all present in 20 to 40 percent of all analyzed sequences. Furthermore, Figures 2(b) and 2(c) depict the length (x-axis) and number (y-axis; log-scale) of patterns found for each dataset for support levels 0.0 - 0.2 and 0.2 - 0.4.

elements in a sequence – i.e., memory effects might be in play (see also Rosvall et al. [25] for a discussion surrounding memory in networks). Consequently, we want to incorporate these potential memory effects into our model approach in the next section, in which we resort to Markov chain models of varying order. The goal is to find a model that can describe action sequences and predict user actions in a sound way.

4. MODELING ACTION SEQUENCES

As our main goal of this work is to predict user actions in collaborative ontology-engineering projects, we need to find an appropriate model that we can fit to the data and leverage for prediction. Our choice falls on Markov chain models which are suitable for modeling categorical sequences. Specific variations of model parameters allow us to incorporate our findings of Section 3; i.e., that regularities and specifically, serial dependence seems to play a role in the action sequences at hand. Consequently, we first give a brief introduction into Markov chain models in Section 4.1 also elaborating a way to incorporate our observations about regularities and patterns in the action sequences. Finally, we will explain two model selection techniques in Section 4.2, which is crucial for deciding between different models, which will help us to evaluate the performance of our models. We then apply the methods established in this section in Section 5.

4.1 Markov Chains

A Markov chain is a stochastic process that models transitions from one state to another based on a given state space S. It usually is referred to as *memoryless* which constitutes the so-called *Markov property* stating that the next state only depends on the current state and not on a series of preceding ones. We now briefly provide an introduction to Markov chains; we point the interested reader to a more thorough introduction in previous work [27, 37].

For such a *first-order* Markov chain[3] – a sequence of random variables $X_1, X_2, \ldots, X_n$ – the following holds:

$$\begin{aligned} P(X_{n+1} = x_{n+1} | X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) &= \\ P(X_{n+1} = x_{n+1} | X_n = x_n) & \quad (1) \end{aligned}$$

Motivated by our observations in Section 3, where we could see that at least some sequences are arranged in a non-random way – i.e., dependence between elements in a sequence – as well as where we could identify longer sequential patterns to be present in our sequences, we are now also interested in extending this notion of memorylessness of Markov chains to also include memory effects. This means, that we not only want to model the next state as being dependent on the current state, but also on a sequence of preceding states (memory effect). Hence, we now also look at Markov chain models of order k where the future depends on the past k states. We can define a Markov chain model of order k as a process that satisfies:

$$\begin{aligned} P(X_{n+1} = x_{n+1} | X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) &= \\ P(X_{n+1} = x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, \ldots, & \\ X_{n-k+1} = x_{n-k+1}) & \quad (2) \end{aligned}$$

Such higher order chains can be modified to a first-order Markov chain by using a state space of compound states of size k[4]; i.e., the state state includes all sequences of length k which finally leads to a set of size $|S|^k|S|$ (see [27] for details). Additionally, we also introduce a so-called *zero-order* Markov chain model where $k = 0$. In such a model the next state does not depend on any other one but we can see this as a *weighted random selection* that should serve as a baseline for our Markov chain models of varying order.

A Markov chain model is represented by a stochastic transition matrix P if the state space is finite (which it is in our case). This matrix contains the transition probabilities of a state x_i to another state x_j for all possible combinations; the probabilities of each row sum to one. The elements of this matrix represent the parameters

[3] For our chains we assume *time-homogeneity*, i.e., the probability of transitions is independent of n.

[4] We prepend k reset states and append one reset state to each sequence so that we "forget" the history of other sequences in the dataset [9].

θ that we have to determine. For doing so we resort to Bayesian inference (see [30, 27] for details). We use a Laplace prior for the inference process – i.e., we set each $\alpha_{ij} = 1$.

4.2 Markov Chain Model Selection

As we are interested in modeling memory in the process, we model the data with a set of models with varying orders k and consequently, have to evaluate the performance of each model leading to a determination of the most appropriate order out of this set. We need to note that lower order models are always nested within higher order ones by definition and hence, higher order models will always fit at least as good as lower order ones. Nonetheless, such higher order Markov chain models need exponentially more parameters and thus may result in severe overfitting.

First, we apply Bayesian model selection [30, 27] giving us a tool to decide between an array of models. The benefit of this method is that it naturally includes a *Occam's razor*, which means that higher order models receive a penalty due too much higher complexity, which can help us to avoid overfitting and give us insights into significance [17].

As a second method for evaluating varying order Markov chain models we use a stratified[5] k-fold cross-fold validation[6]. Following the concepts of Singer et al. [27] and Walk et al. [37] we train the Markov chain models on each training set and validate the predictive power on the test set. First, we rank the probabilities of each row in the transition matrix – which are the expectations of the Bayesian posterior – using *modified competition ranking* that includes a natural *Occam's razor* for higher orders. Next, we determine the rank of each transition of the test set – i.e., from each *start state* to each *target state* – and henceforth, average over all transitions in the test set. Finally, we average over all folds and visualize the results. Note that the best accuracy to be achieved would be one as this would mean that each transition in the test set would be the highest probability of the transition matrix learned from the training set. This method also directly gives us a prediction accuracy of each model that can provide us with insights into the general prediction performance of a model.

5. PREDICTING USER ACTIONS

In this section we present results for fitting and evaluating (via prediction) the Markov chain models of varying order for all conducted analyses (see Section 4.2). We were not able to conduct all analyses for NCIt, as the ontology editor used for developing NCIt exhibits some special functionality, which makes it impossible to extract chronologically ordered change-types and properties (cf. Section 2).

5.1 Predicting Users for Classes

The Bayesian model selections (see Table 2) mostly suggest first- or second-order Markov chain models to be appropriate fits for the underlying data. Only for NCIt a higher order – i.e., a fifth-order – is suggested. In order to study the predictive power of these varying order Markov chain models, we conducted a stratified 3-fold cross-fold validation task (see Figure 3(a) and Table 2) which mostly agrees with our Bayesian model selection results in terms of order appropriateness. This means, that a first- (ICD-11, ICTM and BRO) or second-order (NCIt and OPL) model are shown to have the best predictive power throughout all datasets (accounting for overfitting).

The results indicate that the next event in a sequence seems to be dependent on at least the previous one; partly, also on a sequence of previous states (memory effects). Such Markov chain models (of first or second order) can be used for predicting the next contributor for a class while simultaneously compensating for overfitting. An average position of mostly below two can be achieved with the corresponding best working model.

This tells us that we have a well-working tool for predicting the user that is most likely changing a class next. We may leverage this for recommending classes to users which are eligible for change. By doing so we may manage to severely improve the workflow of users as they may not need to tap into their own intuitions about which class to change next. Also, this process could improve the quality of some classes by automatically finding experts who should edit the class.

5.2 Predicting Change Types for Users

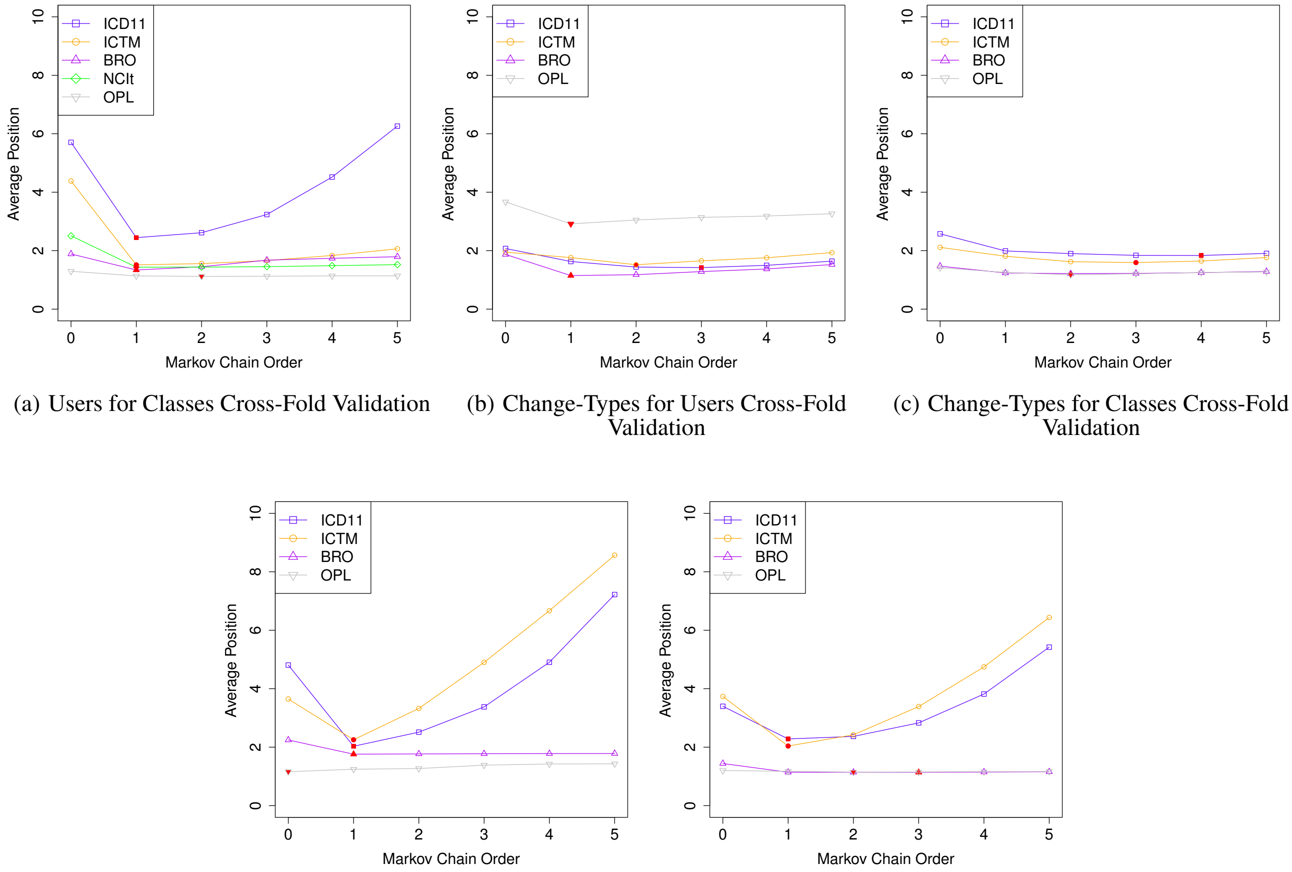
The Bayesian model selection (see Table 2) suggests a fourth-order Markov chain model for ICD-11 and ICTM, a second-order model for BRO and a first-order model for OPL. Subsequently, we conducted a 3-fold stratified cross-fold validation for ICD-11 and ICTM and a 2-fold stratified cross-fold validation for OPL and BRO, due to the smaller number of users available in the latter two datasets (see Figure 3(b) and Table 2). The results suggest that a third-order Markov chain model performed best for predicting the change-type a user is going to perform next for ICD-11. For ICTM and OPL a second-order yielded the best prediction results, while a first-order Markov chain model performed best for BRO. The

Table 2: The results for all datasets and all analyses conducted in Section 5. Rows marked with *CV* indicate the order of the best-performing Markov chain models of our stratified cross-fold validation task (Section 4.2). Rows marked with *Bayes* depict the order of the Markov chain models determined by the Bayesian model selection task (Section 4.2).

		ICD-11	ICTM	NCIt	BRO	OPL
Predicting Users for Classes (**Section 5.1**)	Bayes	2	1	5	2	2
	CV	1	1	2	1	2
Predicting Change Types for Users (**Section 5.2**)	Bayes	4	4	-	2	1
	CV	3	2	-	1	2
Predicting Change Types for Classes (**Section 5.3**)	Bayes	4	3	-	2	2
	CV	4	3	-	2	2
Predicting Properties for Users (**Section 5.4**)	Bayes	2	1	-	3	4
	CV	1	1	-	1	0
Predicting Properties for Classes **(Section 5.5)**	Bayes	2	1	-	3	5
	CV	1	1	-	3	5

[5] Stratified refers to the fact that we try to keep the number of observations equal in each fold.

[6] Note that the number of folds is determined individually for each evaluation due to their stratified nature.



(a) Users for Classes Cross-Fold Validation

(b) Change-Types for Users Cross-Fold Validation

(c) Change-Types for Classes Cross-Fold Validation

(d) Properties for Users Cross-Fold Validation

(e) Properties for Classes Cross-Fold Validation

Figure 3: Results for the *Stratified Cross-Fold Validation* analysis: The plots depict the results of the stratified cross-fold validation for all five datasets for the conducted analyses. The filled elements represent the Markov chain model for each dataset, which achieved the best (lowest) average accuracy (position) score in the prediction tasks. The position score is calculated by determining the position of the next most likely state to occur in a test path given k previous states, where k represents the investigated Markov chain order. Probabilities to select the next most likely state are created using the training set to calculate the transition maps for all datasets and Markov chain orders. The figures show that we can model activity sequences for all of our analyses as first- or higher-order Markov chain models perform best in our prediction task for all datasets, with the only exception of OPL for the *Predicting Properties for Users* analysis (see Figure 3(d)).

cross-fold prediction task also yielded an average accuracy (position) between roughly 1.8 and 3.5.

This indicates that higher-order Markov chains can be used for predicting the change-type a user is most likely to perform next. Practitioners may use this information for recommending change types users should edit next. By doing so we may help to improve the overall progress and quality of the ontology; e.g., if we know that several areas of the ontology or classes lack certain changes, we can steer contributors, which exhibit a preference to perform these kinds of changes, into a specific direction and enforce their contributions in certain branches of the underlying knowledge representation.

5.3 Predicting Change Types for Classes

As depicted in Table 2 the Bayesian model selection suggests a second-order Markov chain model for BRO and OPL, while a third-order model for ICTM and a fourth-order Markov chain model for ICD-11 work best. A stratified 3-fold cross-fold validation (see Figure 3(c) and Table 2) completely agrees with these results for all datasets. The best fitting Markov chain models allow for an average prediction accuracy (position) between 1.8 and 2.0.

The presented results indicate that we can predict the change-type that is most likely conducted on a class next, given at least the two most recent changes on said class as input for our trained Markov chain models. Similar to predicting change types for users, practitioners can use this information for recommending change types that may be useful to change next on a given class. For example, if a class is most likely to receive a certain change type next, we can combine this information with the change types for users and identify a suitable contributor to recommend this class for editing.

5.4 Predicting Properties for Users

The Bayesian model selection yields a second- and first-order Markov chain model for ICD-11 and ICTM and a third- and fourth-order model for BRO and OPL (see Table 2). The conducted 3-fold stratified cross-fold validation, to predict the property a specific user is most likely to change next, yielded a first-order Markov chain model for ICD-11 and ICTM (see Figure 3(d) and Table 2). Due to a limited number of users, a stratified 2-fold cross-fold validation was conducted for BRO and OPL, which showed that a first- and zero-order Markov chain model performs best for predicting the next property for a given user respectively. This means that there was no difference between the Markov chain models trained for OPL and randomly (weighted) choosing (zero-order) the property a user is most likely to change next.

This also means, that for ICD-11, ICTM and BRO we were able to show that subsequent properties users change are dependent on each other; at least for an order of one, which allows for an average prediction accuracy between 1.9 and 2.2. For OPL, the Bayesian model selection and the cross validation approaches do not directly agree with each other; i.e., the Bayesian method suggest an order of four while, interestingly, cross validation would prefer an order of zero (weighted random selection).

In general, by using at least first-order Markov chains it is possible to predict the property a user is most likely to change next for all datasets, except OPL. For steering users into the right direction, we may recommend appropriate properties to change next to contributors.

5.5 Predicting Properties for Classes

Our Bayesian model selection results (see Table 2) suggests for ICD-11 and ICTM a second- and first-order Markov chain model respectively. Furthermore, the results indicate that for BRO a third- and for OPL a fifth-order seem to be appropriate. A stratified 3-fold cross-fold validation (see Figure 3(e) and Table 2) yielded the same results, except for ICD-11, where a first-order model, instead of a second-order model, represents the best predictive accuracy for the underlying data. The conducted cross-fold validation prediction task yielded an accuracy (average position) between roughly 1.8 and 2.4.

Again, our results indicate that we can predict the property that is changed next for a given class reasonably well by using at least a first-order Markov chain. Similar to predicting properties for users, we may now enhance the overall quality of the ontology in an automatic way by aligning the gained information with the properties derived from our user analysis results and recommend users to change specific suitable properties of classes next.

6. SUMMARY AND DISCUSSION

In the previous sections we have studied action sequences of five collaborative ontology-engineering projects from the biomedical domain (see Section 2). To begin with, we provided an initial analysis regarding regularities and sequential patterns in Section 3 to give a basic insight into the processes underlying the user action sequences at hand. First, we started by looking at randomness and regularities by applying an adopted version of the so-called *runs test* exemplary to the ICD-11 dataset in Section 3.1. Our results clearly indicated that a significant array of sequences, based on different features, are produced in a non-random way; this means that at least a portion of sequences is produced in a clustered or systematic arrangement. These observations warranted further studies regarding detailed insights into how these potential regularities look like; hence, we focused on mining sequential patterns next (see Section 3.2). We applied *PrefixSpan* on our User sequences and could identify numerous sequential patterns of longer length – specifically lengths 2 to 4. This lead us to the conclusion that longer patterns seem to play a crucial role in contributor logs of collaborative ontology-engineering projects and that there might be a dependence between subsequent elements in the sequences at hand. Consequently, we hypothesized that it would be beneficial to consider memory effects when modeling our data, and thus user actions. This means, that we wanted to incorporate information of the past into deriving future information – for example, it might be useful to check the two past properties a user has changed for predicting the property she will most likely change next.

For doing so we resorted to Markov chain models of varying order (see Section 4.1) that we applied to our data. We used a Bayesian model selection method for finding the appropriate order for each set of sequences at interest. Supplementary, we were interested in investigating the predictive power of such models, which we evaluated using a cross validation task as described in Section 4.2. The results, as shown in Section 5, confirm our hypotheses: It is indeed useful to incorporate memory effects into the process of modeling user contribution in collaborative ontology-engineering projects. This is particularly imminent as several higher order models are to be preferred throughout all investigations, as can be seen in Table 2. For example, an order of three means that we can best model or predict the next event (e.g., property) by looking at the past three events in a sequence – hence, memory effects are in play. We need to note that all our applied methods compensate the goodness of fit with the corresponding complexity of a model, thus, we penalize higher orders (Occam's razor) which is a necessary step for accounting for potential overfitting.

We can see that both the Bayesian model selection as well as the cross validation prediction task mostly result in similar order suggestion even though they are based on distinct approaches. If the outcome of both methods differ, we can for the most part observe that the cross validation method ensues slightly lower orders than the Bayesian method. This can be explained by the different ways both methods work. The Bayesian method always learns the Markov chain model on the complete model and then performs a model selection strategy which is based on comparing the posterior probabilities of varying order models. Contrary, the cross validation technique learns the Markov chain on a different set (training) compared to where it is evaluated (testing). These differences also account for the drastic mismatch observed between the cross-fold validation prediction task and the Bayesian model selection for OPL in our *Predicting Properties for Users* analysis, where only a very limited number of sequences (three) with unevenly distributed properties across these sequences, is available. Also, the way we rank the probabilities in the cross validation evaluation influences the outcome. Currently, we use modified competition ranking which assigns the worst rank to ties and hence, we very strictly penalize higher orders. Hence, it comes to no surprise for us that if different, the cross validation mostly suggest lower orders than the Bayesian approach. One advantage of the Bayesian approach though is that we could further incorporate penalizations of higher orders when working with model selection; e.g., using an exponential prior [27].

In general, the application of Markov chains on the activity logs of five collaborative ontology-engineering projects has shown that regularities exist. These regularities can potentially be used and exploited by project and community managers to augment and assist users in contributing to the underlying structured knowledge representation. For example, knowing which property a user is most likely to change next and which user is most likely to change a specific concept next could be used to automatically adjust and modify the interface to allow for quicker and personalized workflows. This is especially important for projects the size of ICD-11 or NCIt with thousands of potential classes to contribute to.

We also need to note that the corresponding orders that get suggested might also be – at least to some extent – influenced by how the sequences are shaped; i.e., potential influence factors might be: the distribution of the length of sequences or the number of sequences in a dataset. However, we can argue that these are also properties emerging from how users behave in such systems. Yet, if we are specifically interested in comparing the models of different datasets we need to look deeper into these factors which we leave open for future work. Furthermore, we only work with limited data which also influences the choice of order. Precisely, the number of distinct states as well as the number of observations affect the appropriate order. Basically, the more states one works with, the more difficult it is to compensate the much higher complexity of higher order models with the goodness of fit. Also, we do not necessarily know what would happen if we would perform our investigations on an unlimited number of observations; most likely higher orders will then statistically significantly outperform lower ones (that we e.g., found in our studies) – notwithstanding, working with limited data is a common scenario for researchers and practitioners warranting our experiments and findings.

7. RELATED WORK

The work presented in this paper was inspired by work of the following research areas: Collaborative ontology-engineering, Markov chains and sequential pattern mining.

7.1 Collaborative Ontology Engineering

An ontology represents an explicit specification of a shared conceptualization [14, 5, 32]. In computer-science, this definition usually refers to a construct (formalization) that is automatically processable by a machine representing an abstraction of the real world (shared conceptualization). Ontologies allow computers to "understand" relationships between entities and objects that are modeled in an ontology.

On the other hand, collaborative ontology engineering represents a new field of research with many new problems, risks and challenges. Contributors of such projects, similar to Wikipedia, engage remotely (e.g., via the Internet or a client–server architecture) in the development process to create and maintain an ontology. As mentioned, an ontology represents a formalized and abstract representation of a specific domain; thus, disagreements between authors on certain subjects can occur and tools are needed that augment collaboration and help contributors in reaching consensus when modeling these (and other) topics. Indeed, the majority of the literature about collaborative ontology engineering sets its focus on surveying, finding and defining requirements for the tools used in these projects [20, 13]. Various tools have been developed, specifically aiming at supporting the collaborative development of ontologies. For example, Semantic MediaWikis [18] and its derivatives [2, 12, 26] add semantic, ontology modeling and collaborative features to traditional MediaWiki systems.

Protégé, WebProtégé [34] and its extensions and derivatives for collaborative development are prominent stand-alone tools that are used by a large community worldwide to develop ontologies in a variety of different projects. Both WebProtégé (and its derivatives) and Collaborative Protégé have shown to provide a robust and scalable environment for collaboration and are used in several large-scale projects, including the development of ICD-11 [33].

For analyzing and visualizing the collaborative processes that occur during these projects, Pöschko et al. [24] and Walk et al. [36] have developed *PragmatiX*, a tool that allows to visualize and analyze aspects of the history of collaboratively engineered ontologies. The tool also provides quantitative insights into the ongoing collaborative development processes. Falconer et al. [11] investigated the change-logs of collaborative ontology-engineering projects, showing that users exhibit regularities in their contribution behavior when editing to the ontology. Strohmaier et al. [31] analyzed the collaborative processes in a number of different collaborative ontology-engineering projects by investigating hidden social dynamics and provide new metrics to quantify various aspects of these engineering processes. Wang et al. [39] used association-rule mining to analyze user editing patterns in collaborative ontology-engineering projects.

7.2 Markov chain models

In previous Web studies, Markov chain models have been frequently applied for understanding and modeling Web navigation (e.g., [23, 10, 42]). Mostly, the used Markov chain models were memoryless following the Markovian assumption which is e.g., also modeled in the *random surfer model* in Google's PageRank[8]. Nonetheless, various researchers were also interested in studying the appropriateness of modeling memory effects into models of human navigation – i.e., using higher order chains (e.g., [4, 23]). Yet, the studies revealed that the benefit of higher orders can frequently not compensate the higher complexity and the first-order Markov chain model seems to be a plausible choice. Recently, Chierichetti et al. [9] turned towards again questioning the choice of a first-order chain for modeling human navigation and suggested that the Markovian assumption might be wrong. Consequently, Singer et al. [27] introduced a series of precise model selection techniques for choosing the appropriate Markov chain order. They applied the framework to a series of human navigational datasets and again showed that the memoryless model indeed seems to be a plausible abstraction for human navigation based on the lack of statistically significant improvements of higher order models mostly due to the much higher complexity as already pointed out several years ago. However, the authors also showed that human navigation on a topical level reveals memory effects. Walk et al. [37] adopted this framework to be applicable to structured logs of changes in collaborative ontology-engineering projects and investigated the structure of first-order Markov chains for the change-logs of five different collaborative ontology-engineering projects [38].

7.3 Sequential Pattern Mining

In 1995, Agrawal and Srikant [1] have first addressed the problem of sequential pattern mining. They stated that given a collection of chronologically ordered sequences, sequential pattern mining is about discovering all sequential (chronologically ordered) patterns weighted according to the number of sequences that contain these patterns. The algorithms presented in Agrawal and Srikant [1], in particular AprioriAll and AprioriScale, represent the first *a priori* sequential pattern mining algorithm. In 1996, Srikant and Agrawal [29] further included time-constraints and sliding windows to the definition of sequential patterns and introduced the generalized sequential pattern algorithm (GSP). This means that a specific pattern cannot occur more frequently (above a threshold) if a sub-pattern of this pattern occurs less often (below that threshold). Many other examples of a priori algorithms have been discussed in literature [19, 40, 3], with SPADE [41] being one of the most prominently used and referred to algorithms. One major problem assigned to the a priori based sequential pattern mining algorithms was (in the worst case) the exponential number of candidate generation. To tackle this problem so called pattern-growth approaches have been developed [15, 22].

Many researchers have adapted different algorithms and approaches for different domains to anticipate changing requirements, such as

[16] who analyzed algorithms for sequential pattern mining in the biomedical domain. In Walk et al. [37] the authors have presented a novel application of Markov chains to mine and determine sequential patterns from the structured logs of changes of collaborative ontology-engineering projects.

For the analysis presented in this paper we made use of *PrefixSpan* [22] to investigate if the change-logs of collaborative ontology-engineering projects exhibit commonly used, sequential patterns – we thoroughly introduced this algorithm in Section 3.2.

8. CONCLUSIONS & FUTURE WORK

In this paper our main objective was to predict user actions in collaborative ontology-engineering projects. To that end, we first *explored* if and to what extent regularities and sequential patterns can be extracted from the change-logs of our five datasets. We found that at least a set of sequences were produced in a non-random way and that frequent (longer) patterns can be extracted. We then *modeled* user actions by using Markov chain models which allowed us to incorporate our findings about regularities and patterns. We fitted the models to our sequence data and evaluated them with a specific focus on prediction accuracy. We found that incorporating memory effects (serial dependence) into our models can indeed be useful. The generated predictive models for user actions can not only be used for various recommendation purposes, but also provide project administrators and managers with the means to assess the impact of potential changes on the ontology and the community. For example, knowing which user is most likely to change a specific concept next combined with the information of what kind of change that user is most likely to perform next can potentially be exploited to create personalized task recommendations or to adapt the user-interface to allow for dynamically assisted and faster workflows.

In future work, we first want to extend our choice of models for predicting user action by exploring, for example, varying order Markov chain models, Hidden Markov chain models or Semi Markov chain models. When fitting these models to the data, we plan on providing further evaluation comparisons between these distinct models and consequently, also want to explore the potential of incorporating memory into these alternative models. Furthermore, we want to look at other data sources (e.g., Semantic MediaWikis) to be able to produce more general statements, independent from the datasource, and also closely investigate the influence of different data properties as discussed in Section 6.

We strongly believe that the analysis and predictive models presented in this paper represents an important step towards a better understanding of collaborative ontology-engineering projects in the biomedical domain.

Acknowledgements

We want to thank Natalya F. Noy, Tania Tudorache and Mark A. Musen from the Stanford Center for Biomedical Informatics Research (BMIR) for providing access to the data. This work was partially funded by the DFG German Science Fund grant STR 1191/3-2.

9. REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society.

[2] S. Auer, S. Dietzold, and T. Riechert. Ontowiki - a tool for social, semantic collaboration. In *Proceedings of the 5th International Conference on The Semantic Web*, ISWC'06, pages 736–749, Berlin, Heidelberg, 2006. Springer-Verlag.

[3] C. Bettini, X. S. Wang, and S. Jajodia. Testing complex temporal relationships involving multiple granularities and its application to data mining (extended abstract). In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '96, pages 68–78, New York, NY, USA, 1996. ACM.

[4] J. Borges and M. Levene. Data mining of user navigation patterns. In *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, WEBKDD '99, pages 92–111, London, UK, UK, 2000. Springer-Verlag.

[5] W. Borst. Construction of engineering ontologies for knowledge sharing and reuse. 1997.

[6] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: forecasting and control*. John Wiley & Sons, 2013.

[7] J. V. Bradley. Distribution-free statistical tests. 1968.

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[9] F. Chierichetti, R. Kumar, P. Raghavan, and T. Sarlos. Are web users really markovian? In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 609–618, New York, NY, USA, 2012. ACM.

[10] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Trans. Internet Technol.*, 4(2):163–184, May 2004.

[11] S. Falconer, T. Tudorache, and N. F. Noy. An analysis of collaborative patterns in large-scale ontology development projects. In *Proceedings of the sixth international conference on Knowledge capture*, K-CAP '11, pages 25–32. ACM, 2011.

[12] C. Ghidini, B. Kump, S. Lindstaedt, N. Mahbub, V. Pammer, M. Rospocher, and L. Serafini. Moki: The enterprise modelling wiki. In *The Semantic Web: Research and Applications*, pages 831–835. Springer, 2009.

[13] T. Groza, T. Tudorache, and M. Dumontier. Commentary: State of the art and open challenges in community-driven knowledge curation. *Journal of Biomedical Informatics*, 46(1):1–4, Feb. 2013.

[14] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[15] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 1–12, New York, NY, USA, 2000. ACM.

[16] C.-M. Hsu, C.-Y. Chen, B.-J. Liu, C.-C. Huang, M.-H. Laio, C.-C. Lin, and T.-L. Wu. Identification of hot regions in protein-protein interactions by sequential pattern mining. *BMC bioinformatics*, 8(Suppl 5):S8, 2007.

[17] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[18] M. Krötzsch, D. Vrandečić, and M. Völkel. Semantic mediawiki. In *The Semantic Web-ISWC 2006*, pages 935–942. Springer, 2006.

[19] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.

[20] N. F. Noy and T. Tudorache. Collaborative ontology development on the (semantic) web. In *AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering*, pages 63–68. AAAI, 2008.

[21] P. C. O'Brien and P. J. Dyck. A runs test based on run lengths. *Biometrics*, pages 237–244, 1985.

[22] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th International Conference on Data Engineering*, ICDE '01, pages 215–224, Washington, DC, USA, 2001. IEEE Computer Society.

[23] P. L. T. Pirolli and J. E. Pitkow. Distributions of surfers' paths through the world wide web: Empirical characterizations. *World Wide Web*, 2(1-2):29–45, Jan 1999.

[24] J. Pöschko, M. Strohmaier, T. Tudorache, N. F. Noy, and M. A. Musen. Pragmatic analysis of crowd-based knowledge production systems with iCAT Analytics: Visualizing changes to the ICD-11 ontology. In *Proceedings of the AAAI Spring Symposium 2012: Wisdom of the Crowd*, 2012.

[25] M. Rosvall, A. V. Esquivel, A. Lancichinetti, J. D. West, and R. Lambiotte. Memory in network flows and its effects on spreading dynamics and community detection. *Nature Communications*, 5, 2014.

[26] T. Schandl and A. Blumauer. Poolparty: SKOS thesaurus management utilizing linked data. *The Semantic Web: Research and Applications*, 6089:421–425, 2010.

[27] P. Singer, D. Helic, B. Taraghi, and M. Strohmaier. Detecting memory and structure in human navigation patterns using markov chain models of varying order. *PloS one*, 9(7):e102070, 2014.

[28] N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W.-L. Shaiu, and L. W. Wright. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, February 2007.

[29] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, UK, 1996. Springer-Verlag.

[30] C. C. Strelioff, J. P. Crutchfield, and A. W. Hübler. Inferring markov chains: Bayesian estimation, model comparison, entropy rate, and out-of-class modeling. *Physical Review E*, 76:011106, Jul 2007.

[31] M. Strohmaier, S. Walk, J. Pöschko, D. Lamprecht, T. Tudorache, C. Nyulas, M. A. Musen, and N. F. Noy. How ontologies are made: Studying the hidden social dynamics behind collaborative ontology engineering projects. *Web Semantics: Science, Services and Agents on the World Wide Web*, 20(0), 2013.

[32] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. volume 25, pages 161–197, 1998.

[33] T. Tudorache, S. Falconer, C. Nyulas, N. F. Noy, and M. A. Musen. Will semantic web technologies work for the development of icd-11? In *The Semantic Web–ISWC 2010*, pages 257–272. Springer, 2010.

[34] T. Tudorache, C. Nyulas, N. F. Noy, and M. A. Musen. WebProtégé: A Distributed Ontology Editor and Knowledge Acquisition Tool for the Web. *Semantic Web Journal*, 4(1/2013):89–99, 2013.

[35] A. Wald and J. Wolfowitz. On a test whether two samples are from the same population. *The Annals of Mathematical Statistics*, 11(2):147–162, 1940.

[36] S. Walk, J. Pöschko, M. Strohmaier, K. Andrews, T. Tudorache, C. Nyulas, N. F. Noy, and M. A. Musen. PragmatiX: An Interactive Tool for Visualizing the Creation Process Behind Collaboratively Engineered Ontologies. *International Journal on Semantic Web and Information Systems*, 9(1):45–78, 2013.

[37] S. Walk, P. Singer, M. Strohmaier, D. Helic, N. F. Noy, and M. A. Musen. Sequential usage patterns in collaborative ontology-engineering projects. *arXiv preprint arXiv:1403.1070*, 2014.

[38] S. Walk, P. Singer, M. Strohmaier, T. Tudorache, M. A. Musen, and N. F. Noy. Discovering beaten paths in collaborative ontology-engineering projects. *Journal of Biomedical Informatics*, 2014.

[39] H. Wang, T. Tudorache, D. Dou, N. F. Noy, and M. A. Musen. Analysis of user editing patterns in ontology development projects. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, OTM '13, pages 470–487. Springer, 2013.

[40] J. T.-L. Wang, G.-W. Chirn, T. G. Marr, B. Shapiro, D. Shasha, and K. Zhang. Combinatorial pattern discovery for scientific data: Some preliminary results. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, SIGMOD '94, pages 115–125, New York, NY, USA, 1994. ACM.

[41] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2):31–60, 2001.

[42] I. Zukerman, D. W. Albrecht, and A. E. Nicholson. *Predicting users' requests on the WWW*. Springer, 1999.