An Empirical Evaluation of Property Recommender Systems for Wikidata and Collaborative Knowledge Bases

Eva Zangerle, Wolfgang Gassler, Martin Pichl, Stefan Steinhauser, Günther Specht
Databases and Information Systems
Department of Computer Science
University of Innsbruck, Austria
{firstname.lastname}@uibk.ac.at

ABSTRACT

The Wikidata platform is a crowdsourced, structured knowledgebase aiming to provide integrated, free and language-agnostic facts which are—amongst others—used by Wikipedias. Users who actively enter, review and revise data on Wikidata are assisted by a property suggesting system which provides users with properties that might also be applicable to a given item. We argue that evaluating and subsequently improving this recommendation mechanism and hence, assisting users, can directly contribute to an even more integrated, consistent and extensive knowledge base serving a huge variety of applications. However, the quality and usefulness of such recommendations has not been evaluated yet. In this work, we provide the first evaluation of different approaches aiming to provide users with property recommendations in the process of curating information on Wikidata. We compare the approach currently facilitated on Wikidata with two state-of-the-art recommendation approaches stemming from the field of RDF recommender systems and collaborative information systems. Further, we also evaluate hybrid recommender systems combining these approaches. Our evaluations show that the current recommendation algorithm works well in regards to recall and precision, reaching a recall@7 of 79.71% and a precision@7 of 27.97%. We also find that generally, incorporating contextual as well as classifying information into the computation of property recommendations can further improve its performance significantly.

Keywords

Wikidata, Wikipedia, Recommender Systems, Evaluation

1. INTRODUCTION

Within the last years, the Wikidata platform has evolved to a central, consistent and structured knowledge base, which is maintained by an active community. The knowledge captured is provided in both a human- and machine-readable format (due to its structured nature) and can be used across different languages as stored facts are not tied to a single language [21]. The data provided is facilitated by an increasing number of applications with Wikipedia being the most popular one. As of March 2016, Wikidata features more than 17 million data items, which in principle "represent all the things in human knowledge, including topics, concepts, and objects"[1]. E.g., Albert Einstein or the city of New York are such items. On Wikidata, these items are described by so-called statements, where a property serves as a descriptor for a value representing the actual information. E.g., a property of the item Albert Einstein is dateOfBirth and the according value is 1879-03-14. Wikidata items and the according statements have been curated by approximately 315 million edits[2]. On average, each item on the platform is shaped by 14 edits performed by either by humans or bots.

As of 2014, 90% of all edits were made by bots as the Wikidata project strives to automate tasks and outsources these to bots [23]. However, still roughly one million edits are performed by human users every month and hence, contributions by human users play an important role. To improve this manual process of inserting new facts, the Wikidata platform supports its committed community by a so-called "property suggestor". This tool assists the user in entering information by providing suggestions for novel properties which are likely to be added to the current data item. The computation of such suggestions is based on association rules [3] and hence, on the co-occurrence of properties on existing Wikidata items. The current algorithm further considers so-called "classified properties"—currently "instanceOf" or "subclassOf"—which are further exploited to derive information about the type of the item described. However, to this end, the usefulness and quality of such suggestions have not been thoroughly evaluated yet.

In this work, we aim to get a deeper understanding for such recommender systems, to which extent the user is supported and particularly, how the individual evaluated algorithms differ in terms of recommendation accuracy and suitability in the context of a user. Therefore, our research is driven by the following research questions (RQ):

- **RQ1:** How do the presented property recommender system algorithms perform in terms of recommendation accuracy?
- **RQ2:** How can we improve the existing predicate recommender systems for Wikidata?

[1] https://www.wikidata.org/wiki/Help:Items
[2] https://www.wikidata.org/wiki/Special:Statistics

OpenSym 2016 Berlin, Germany

DOI: 10.475/123_4

The main contribution of this study is that it is the first detailed study on recommender systems assisting users in entering information on the Wikidata platform. We provide a detailed description, analysis and comparison of a set of recommendation approaches. Moreover, we propose a general evaluation framework which allows for also comparing future approaches (available on Github[3]). We find that the current implementation of the Wikidata Entity Suggester provides property recommendations of higher accuracy than the other proposed approaches. During the course of our evaluation, we identify two key aspects which are crucial for the quality of recommendations: incorporating classifying properties and making use of contextual information for ranking the property recommendation candidates. Therefore, we propose to merge the best performing aspects of each of the three recommendation approaches by adding contextual information and information about classifying properties. Our evaluations show that a hybrid approach combining the current Wikidata Entity Suggester recommendation engine with a context-based ranking computation as proposed by the Snoopy system [7] improves recall as well as precision of the recommendations significantly.

The remainder of this paper is structured as follows. Section 2 presents background and works related to the presented evaluation. Section 3 presents the recommender systems which we evaluate in the course of this study. Section 4 describes the methods and data set facilitated to perform the evaluation. Section 5 describes the results of the performed evaluation which are further discussed in Section 6. Section 7 concludes the paper.

2. BACKGROUND AND RELATED WORK

In literature, we can find two different fields of research relevant for the conducted study: (i) recommendation approaches for triples and particularly, Wikidata information, and (ii) research investigating various aspects regarding the Wikidata platform.

Naturally, approaches dealing with recommending data on the Wikidata platform are closely related to the conducted study. Therefore, we aim to compare the approach currently facilitated on the Wikidata platform with further approaches. We find that the so-called Predicate Suggestion approach [2] and the Snoopy approach [25] aim to suggest predicates in a triple-based environment. Therefore, these are also applicable in the Wikidata setting and hence, included in the evaluation. As these algorithms are essential for the presented study, we discuss these in more detail in Section 3.

Recently, the Wikidata platform has attracted much attention in the research community. This naturally includes describing and promoting the general idea behind Wikidata [23, 22]. Likewise, Wikidata has also been investigated in regards to its data model, its integration into the Semantic Web by exporting parts of the Wikidata project as RDF [6]. Similarly, Hernandez et al. perform an analysis of different reification approaches aiming to transform Wikidata knowledge to RDF to be able to integrate it into Semantic Web use cases and applications [9]. Müller-Birn et al. provide a thorough analysis on whether edits and contributions on the Wikidata platform are similar to Wikipedia, and hence, Wikidata resembles a peer-production system or whether edits can rather be seen as collaborative ontology engineering effort [12]. The authors come to the conclusion that driven by the fact that Wikidata is currently still in a data-gathering-phase, contributions resemble a peer-production system. Ta et al. propose an alignment approach for Wikipedia infoboxes and Wikidata to complement existing Wikipedia knowledge by introducing new interwiki-links across individual Wikipedias of different languages [19]. Steiner et al. propose a platform aiming to analyze edits on both Wikipedia and Wikidata and provide detailed analysis on the distribution of edits among those two platforms as well as the number of human editors and bot editors [18]. Also, Wikidata has lately been increasingly facilitated to improve data quality on Wikipedia in particular topics such as e.g., drug-drug interaction [13] or data about human and mouse genes [4].

[3]https://github.com/dbisibk/PropertyRecommenderEvaluator

3. RECOMMENDER ALGORITHMS

The following section describes the recommender algorithms that are compared in our evaluation. As already depicted in the previous section, there currently exist three different approaches which allow for recommending Wikidata items: (i) the recommender system as provided by Wikidata itself, (ii) the Predicate Suggestion approach for the autoammendment for triples and (iii) the Snoopy approach. In the following, we firstly shortly introduce association rules as the basis of all three approaches. Secondly, we sketch the strategies the three recommender systems take.

One aspect all of the presented recommendation algorithms have in common is that these all are based on association rules (AR) [3]. The main goal of association rule learning is to find co-occurring items within a database or from a more general perspective: to find patterns within data. Initially, AR were used to find items commonly appearing together in shopping baskets, aiming to provide a strong foundation for marketing decisions. Later, AR have been applied in various different settings, such as network intrusion detection [20] or credit card fraud detection [14]. Also, AR have been facilitated as the basis for recommender systems, e.g., in [16, 24]. In principle, association rules are based on a set of items (in the Wikidata case, properties) and a set of transactions (in the Wikidata case, the set of all data items and the according properties). An association rule is then defined as an implication expression $X \rightarrow Y$, where X and Y are item sets. Such a rule describes that there is a relation between X and Y, particularly, that the properties X and Y appear on the same data items. In the case of the presented recommender algorithms, due to performance reasons both the head (antecedent) and the tail (consequent) of the rules are single items (in contrast to traditional association rules, where these may also be sets of items). By taking the properties used on a given data item as input, such rules allow for computing recommendations by extracting all rules featuring any of the input properties in the rule's antecedent and subsequently deriving recommendations by ranking the properties featured in the consequences of these rules.

Generally, AR are precomputed before the recommendation process and only updated in the case of changes, i.e., the addition of new properties to data items. Regarding the creation of rules, the Apriori-algorithm for the computation of AR [3] has paved way for a set of optimizations regarding the computational effort for the rule computations as e.g., the FP-Growth algorithm [8].

In the field of association rule mining, often a lower bound is defined for the creation of such rules in terms of the popularity of rules (support) or the confidence of rules. I.e., we only create rules based on the given data set if these fulfill these lower bounds. However, in the performed evaluation, these measures are solely utilized for ranking recommendations. The support of a rule in the Wikidata case can be defined as stated in Equation 1.

$$support(X \rightarrow Y) = \frac{|\{t \in D | (X \cup Y) \subseteq t\}|}{|D|} \quad (1)$$

where D is the set of all transactions in the data set (in the Wikidata case, data items). I.e., the support defines the fraction of transactions which actually contain both the antecedent and the consequent of the rule.

The confidence of an AR can be defined as stated in Equation 2. This measure can be interpreted as the conditional probability of Y appearing on the same data item given that X is a property contained on the data item.

$$conf(X \rightarrow Y) = \frac{support(X \cup Y)}{support(X)} \quad (2)$$

Based on this definition of association rules, we now present the evaluated recommender algorithms.

3.1 Predicate Suggestion

Abedjan and Naumann propose an approach aiming to enrich RDF data sets by a set of rule mining approaches [1, 2]. This approach is also inspired by traditional Association Rules [3]. Among those approaches, Abedjan and Naumann claim that once RDF triples (consisting of subject, property and object) are grouped by their subject, traditional association rules can be facilitated to provide predicate suggestions to the user. Based on the previously inserted predicates for a given subjects, this approach computes predicates to be suggested based on all rules that incorporate these predicates as antecedents of their rules. Consequently, the predicates appearing in the consequences of those rules are extracted and form the set of possibly suggested predicates. These candidate predicates are subsequently ranked by the sum of the confidence values of all rules that actually have the corresponding predicate as their consequence. In the remainder of this paper, we will refer to this approach as *AN* (for Abedjan and Naumann).

3.2 Wikidata Property Suggester

The Wikidata platform provides users with a property suggester which aims at assisting users when adding new statements to a given Wikidata item. The code underlying the suggester can be found in the project's GitHub repository[4]. The Property Suggester can also be accessed using the Wikimedia API[5].

The approach taken by the Entity Suggester is based on the Predicate Suggestion approach by Abedjan and Naumann as described in the previous section. However, the Wikidata Property Suggester further adds so-called "classifying" properties, which are the properties "instanceOf" and "subclassOf". The property instanceOf refers to the fact that the described item "is a specific example and a member of that class"[6], whereas the subclassOf property signifies that all instances of these items are instances of the given superclass[7]. For the recommendation approach, these properties are treated differently as for these, not only the property is used as a antecedent of rules, but the combination of the property and each occurring object. I.e., $(instanceOf, human) \rightarrow dateOfBirth$ rules are formed and added to the set of rules. This allows to add further information to the recommendation computation process as information about the type of the given data item can be inferred and exploited (if available). Naturally, such information is valuable for the recommendation of suitable properties. We further refer to this approach as *WD* (for WikiData).

3.3 Snoopy Property Recommendation

The so-called Snoopy-concept is in principle a collaborative information system which allows users to store information in the form of triples [25, 7]. The system's main contribution is that it assists users when entering information by providing recommendations for properties, objects as well as links between different subjects. These measures aim to (i) create a common schema among all subjects to without restricting the user in entering information and (ii) encourage the user to add more information as basically, the user may enter further information by simply accepting the proposed properties and entering the according values (objects). One aspect in assisting users in entering information on the Snoopy platform is the recommendation of properties which may be suitable for a given topic. Hence, Snoopy provides a recommender system compatible with the Wikidata Property Suggester. Snoopy's recommendations are computed based on association rules, where recommended properties are ranked by the number of occurrences of a given rule across all data items (in contrast to the sum of confidences in the Abedjan and Naumann approach). We refer to this approach as *SN* (for Snoopy).

An extension to the SN-approach is to use contextual information for ranking properties. In this case, the notion of context refers to the rules' antecedents as these describe the context a given property is embedded in. I.e., the set of properties co-occurring with a given property provide contextual information about the property. The more such information about a property is available, the broader the foundation for a recommendation of the property. Therefore, we utilize the number of different contexts a rule is embedded in for the ranking computation. I.e., we count the number of distinct rules that actually have a given property as its consequence. Such an approach has already been facilitated by Sigurbjörnsson et al. [17], who refer to this approach as a voting approach. For this context-based approach, we rank properties by the number of distinct rules leading to the property. I.e., the higher the number of distinct rules with a given property in the consequence, the higher the rank of the property. In case of a rank tie, we further rely on the number of total occurrences of the particular rule to resolve the tie. We refer to this approach as *SN_context*.

3.4 Hybrid Recommender Algorithms

We further propose to evaluate hybrid variants of the three recommender approaches presented previously. We identify

[4]https://github.com/Wikidata-lib/PropertySuggester

[5]https://www.wikidata.org/w/api.php?action=help&recursivesubmodules=1#wbsgetsuggestions

[6]https://www.wikidata.org/wiki/Property:P31

[7]https://www.wikidata.org/wiki/Property:P279

one distinctive characteristic for each of the presented approaches: (i) using the sum of the rule confidences of all applicable rules for ranking in case of the AN-approach, (ii) utilizing classifying properties to add type information in the case of the WD-approach and (iii) the use of context–information for ranking of property candidates in case of the SN-approach.

We propose to evaluate each of the approaches extended by the distinctive characteristics of the other two recommendation algorithms presented. This includes combining the SN_simple approach with classified properties as described in Section 3.2 (*SN_classified*), as well as adding context to perform the ranking in this hybrid configuration (*SN_context_classified*). A second stream of hybrid configurations results from taking the Predicate Suggestion approach as described in Section 3.1 and combining it with Snoopy's contextual ranking approach (*AN_context*) and lastly, we also propose to extend the WD approach with Snoopy's context ranking approach (*WD_context*).

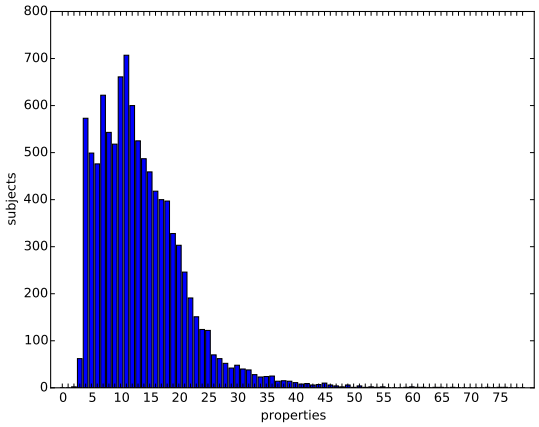


Figure 1: Test Set Properties per Item Distribution

4. METHODS AND DATA

In the following section, we present the data underlying the conducted evaluation and the methods used to ensure a valid and fair comparison of the recommender algorithms presented in Section 3.

4.1 Data Set

For the evaluation of recommender systems aiming to provide users with suitable recommendations during the process of entering information in the Wikidata system, we naturally rely on the Wikidata data set. Therefore, we gathered the full database dump in JSON-format from the website[8], using the version of 2015-10-26. In a second step, we imported the full Wikidata data set into a MariaDB using a triple format [7] to ensure a fully integrated view on the data, which subsequently forms the basis for all evaluations of the recommender systems. All of the evaluated approaches are originally implemented based on a rule-database, therefore we rely on this design choice and base the evaluation on such a database (or rather, multiple configurations of it) for the computation of the recommendations. I.e., we compute all rules for the given configurations and store these in the database and do not consider any support or confidence threshold for the rule creation. The subsequent evaluations rely on these rules only.

Characteristic	Value
Items	18,699,771
Distinct objects	23,097,836
Distinct properties	1,771
Avg. properties per item	4.13
Min. properties per item	1
Max. properties per item	939

Table 1: Wikidata Data Set Characteristics

Table 1 lists the most important characteristics of the data set. As can be seen, the data set contains facts about approximately 18 million items, which are described by 1,771 distinct properties and 23 million distinct objects. On average, each item is described by 4.13 property-value pairs, the minimum number of properties per item is 1, the maximum number is 913. The low mean value can be explained by the fact that 7,608,752 items are described by a single property-value pair, 1,743,294 items are described by two property-value pairs. I.e, 50.01% of all items feature at most two property-value pairs.

4.2 Evaluation Setup

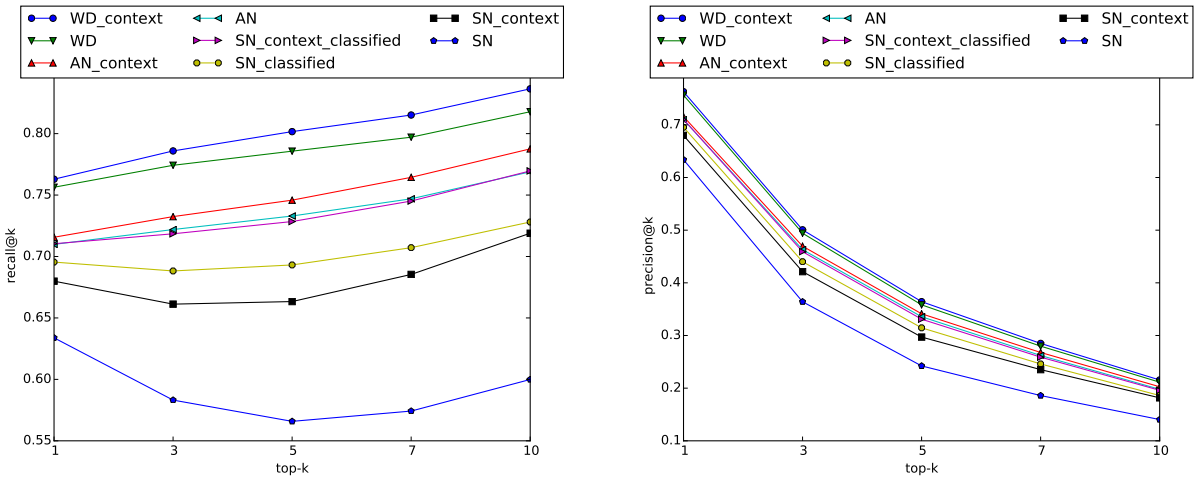
In the following section, we present the evaluation setup facilitated for evaluating and comparing the different recommendation approaches. The code is available on GitHub[9]. Along the lines of previous studies on recommender systems for Wikipedia infobox data [25]—which we consider highly similar to the evaluation at hand—we evaluate the recommender algorithms presented in Section 3. We employ a leave-n-out strategy [5] to evaluate their ability to provide suitable recommendations based on the input data as well as for their ability to reconstruct the triples of a given item.

Based on the data set as described in Section 4.1, we employ the following evaluation strategy. First, we randomly select 10,000 different data items that form the test set for the evaluation. Our minimal requirement in regards to the size of the item in terms of the number of properties is that each item in the test set contains at least four properties. This requirement is fulfilled by 9,254 data items that subsequently form the test set underlying the evaluation. Figure 1 shows the distribution of the number of properties on each item for the test set.

For each of these data items within the test set, we randomly select three properties and remove all but these three properties from the data item. We store these removed properties as these form the ground truth data for the evaluation. The resulting reduced data item serves as the first input for the recommender systems. Subsequently, the evaluation process attempts to reconstruct an item by iteratively computing recommendations for the item. If the current list of property recommendations contains one or more properties which have previously been removed, we take the first of these properties, add it to the data item and recompute

[8]https://www.wikidata.org/wiki/Wikidata:Database_download

[9]https://github.com/dbisibk/PropertyRecommenderEvaluator



(a) Recall@k

(b) Precision@k

Figure 2: Evaluation Measures

the recommendations with the extended, new input item. We repeat this procedure until no more matching property recommendations can be found.

As for the measures for evaluating the reconstruction process we rely on the traditional information retrieval measures recall, precision and F-measure [11]. This allows for evaluating the quality of the provided recommendations. Particularly, we compute these measures for different lengths of recommendation lists to be able to evaluate the performance of the algorithms with varying number of recommendations to be provided. For each of the evaluation runs, we compute the measures for each data item processed and provide the mean of each measure across all data items.

We compute the recall and precision measures as stated in Equation 3 and 4. Please note the slight modification of the recall measure which allows us to reflect on data items where the number of remaining correct properties is lower than the number of recommendations provided—due to the fact that the number of removed properties is constantly decreased and therefore, may be smaller than the number of computed recommendations. I.e., if only two properties have not been reconstructed yet and we provide 10 property recommendations, the recall would naturally be capped at 20%, which we avoid by modifying the measure as stated in Equation 4, where $\mathcal{P}_{rem}$ are the previously removed properties and $\mathcal{P}_{rec}$ is the set of top-k recommendations.

$$precision(\mathcal{P}_{rec}) = \frac{|\mathcal{P}_{rec} \cap \mathcal{P}_{rem}|}{|\mathcal{P}_{rec}|} \tag{3}$$

$$recall(\mathcal{P}_{rec}) = \frac{|\mathcal{P}_{rec} \cap \mathcal{P}_{rem}|}{min(|\mathcal{P}_{rem}|, k)} \tag{4}$$

Furthermore, the F-measure, which combines recall and precision values is computed for our evaluations. The balanced F-measure F_1 is the harmonic mean of recall and precision is defined as stated in Equation 5.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5}$$

To also evaluate the ranking order in which the individual elements are recommended, we make use of the mean reciprocal rank (MRR) measure [11]. The MRR considers the position of correct recommendations and therefore, algorithms providing more correct recommendations at the top positions of the list of recommendations are evaluated to perform better. The mean reciprocal rank measure is depicted in Equation 6, where $pos(item)$ refers to the position of the evaluated property within the list of recommendations.

$$MRR = \sum_{i=1}^{|correct|} \frac{1}{pos(correct_i)} \tag{6}$$

Moreover, we make use of the reconstruction-measure, which describes the percentage of reconstructed properties during the iterative process as proposed in [25]. The measure is depicted in Equation 7, where *reconstructed properties* refers to the number of properties which have been correctly recommended and (virtually) accepted during the iterative evaluation and recommendation process as described previously.

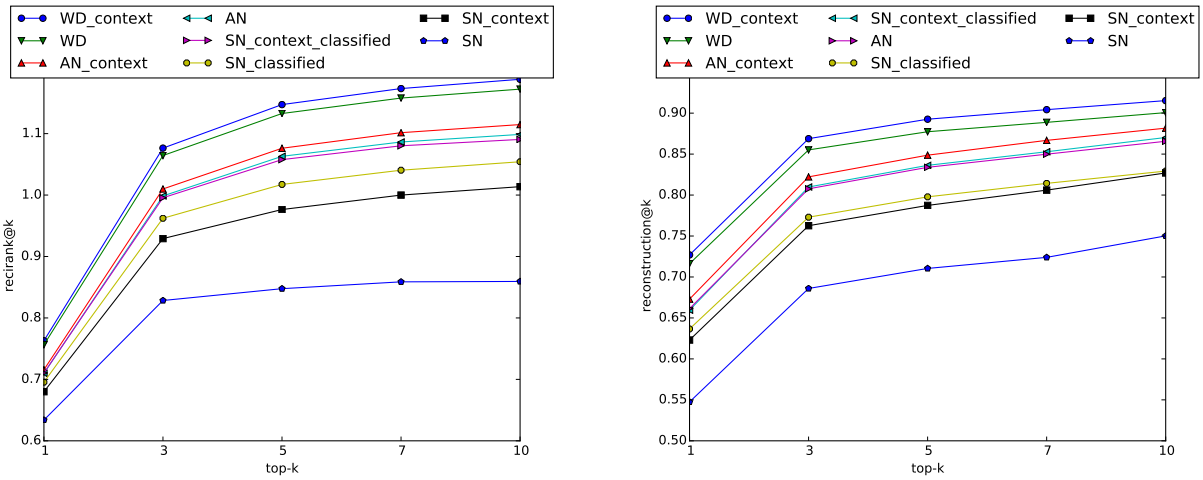
$$reconstruction(item) = \frac{|reconstructed\ properties|}{|removed\ properties|} \tag{7}$$

For the empirical comparison of the individual approaches using these measures, we rely on a Mann Whitney U–test [10] due to the non-normal distribution of our data.

5. RESULTS

This section presents the results of the evaluation methods proposed in the previous section in the light of the posed research questions. Firstly, we present the results of the evaluation of the individual recommendation approaches in regards to recall and precision. Secondly, we get a closer look at the mean reciprocal rank and in a third step, we look at the reconstructive power of the individual algorithms.

Firstly, we evaluate the presented recommender algorithms in regards to recall and precision. Figure 2a shows a plot of the recall@k-measure for the presented recommenders. I.e., we evaluate the recall-measure for 1 to 10 provided recommendations and depict the different results. We observe that



(a) Mean Reciprocal Rank

(b) Reconstruction

Figure 3: Evaluation Measures

in terms of recall, the Wikidata recommendation approach enhanced with contextual information (WD_context as presented in Section 3.3) performs best, achieving a recall@1 of 76.29% and a recall@10 of 83.64%. The evaluations shows that this approach performs best across all numbers of recommendations given (significantly better than all other approaches; $p < 0.001$). The second best approach is the AN approach, followed by SN_context. We can also observe that the SN approach performs last, however, can be significantly improved by utilizing contextual information ($p < 0.001$).

Figure 2b shows the precision@k-results for the evaluated algorithms. Again, WD_context performs best, reaching a precision of 76.29%@1 and a precision of 21.55%@10, again closely followed by the AN approach. We detect significant differences in the performance of these two ($p < 0.001$) and can observe that the algorithms perform rather similar, again with SN being the worst performing algorithm. The evaluations show that using contextual information for ranking improves each of the proposed approaches significantly. We verify this finding by comparing the results of the individual algorithms once without using context for ranking and once with added context information for the ranking computation. This evaluation shows that for all of the approaches, recall as well as precision for all recommendation list sizes is significantly increased when introducing context to the ranking process ($p < 0.001$; Mann–Whitney U test). As for the contribution of classifying properties, we observe similar results. Adding special rules for classifying properties to the set of association rules significantly increases recall as well as precision of all of the approaches ($p < 0.001$).

Figure 3a shows the mean reciprocal rank for all algorithms. This measure can be very helpful when analyzing the performance of the ranking function of a given recommender system. Generally, we observe that the findings in this evaluation are in line with the previous results. I.e., the best performing algorithms in terms of recall and precision also perform well in terms of ranking.

The results of the evaluation of the reconstructive power of the evaluated property recommendation algorithms can be seen in Figure 3b. Please note that these results resemble the reconstruction value at the end of the iterative evaluation. I.e., we count the number of reconstructed properties once no more matching property recommendations are provided by the algorithm. Again, the results correlate with the previous findings and show the best performance for the WD_context approach, significantly better than the WD approach ($p < 0.001$).

To conclude the evaluation of the individual recommender algorithms, we provide a detailed comparison of the best algorithm of all three proposed approaches in Table 2. On the Wikidata platform, the default number of recommendations provided to a user entering new information, is 7. Therefore, we list the performance@7 in Table 2 where we measure performance in terms of recall, precision, the F1-measure, the mean reciprocal rank (MRR) as well as the reconstruction value. As can be seen, the best results are obtained by the WD_context approach across all measures.

To answer RQ1 regarding the quality of the current recommendation algorithm on Wikidata, we can observe that it performs well and the evaluations show that incorporating classified properties and the according objects into the recommendation computation process is a reasonable approach. As for possible improvements to the current algorithm regarding RQ2, we suggest to incorporate contextual information (as proposed by the Snoopy approach) into the recommendation process. The evaluations show that not only considering the sum of confidences of the applicable rules, but also the number of rules leading to a given recommendation candidate leads to significantly better results in terms of recall, precision, MRR and reconstruction.

6. DISCUSSION

In the following section we further discuss the findings of the conducted evaluation.

The performed evaluations reveal two important influence factors when it comes to recommending properties to users who currently enter information on the Wikidata platform: context and classified properties. The evaluations show that incorporating these two aspects into the recommendation process can significantly enhance and improve the resulting recommendations. I.e., using classifying properties to gain

Algorithm	Precision	Recall	F1	MRR	Reconstruction
WD_context	28.49%	81.52%	42.23%	1.17	90.42%
WD	27.97%	79.71%	41.41%	1.16	88.87%
AN_context	26.78%	76.44%	39.66%	1.10	86.67%
AN	26.20%	74.69%	38.79%	1.09	84.98%
SN_context_classified	25.89%	74.51%	38.43%	1.08	85.29%
SN_classified	24.60%	70.71%	36.50%	1.04	81.43%
SN_context	23.52%	68.54%	35.02%	0.99	80.60%
SN	18.59%	57.42%	28.09%	0.86	72.39%

Table 2: Detailed Evaluation of all Algorithms@7

further information about the type of the given data item is an important factor. Similarly, considering the context of a property, and hence, the number of distinct rules leading to its recommendation, for ranking, further adds to high-quality recommendations.

One limitation of the classifying approach we observe lies in its reduced flexibility and hence, its generalizability. We argue that information about the type or superclass of a data item may not always be available, especially when applying these concepts in a broader context. The manual choice of classifying properties seems rather inflexible and is not necessarily generalizable. Therefore, we argue that by setting up a more rigid recommender system by manually specifying classifying properties we trade in flexibility for (slightly) improved results. The evaluations show that e.g., in terms of recall@7, the difference between the WD_context and WD approaches is 1.81%. On the contrary, especially considering the fact that a majority of data items feature only a low number of properties (on average, 4.13 for the data set underlying our evaluations), any recommender system has to face the so-called cold-start-problem [15], which fundamentally describes the lack of data within a recommender system. This cold-start problem could (at least partly) be avoided by exploiting type information as in Wikidata's classifying approach.

With respect to the evaluation procedure, are well aware of the fact that performing a quantitative evaluation as presented in this study poses a limitation. That is, as the goal of a quantitative leave-out analysis [5] is to analyze to which extent recommender algorithms are able to reconstruct a given item. I.e., we evaluate how many of the items that are already utilized on a given data item and were removed for the evaluation were actually recommended. We argue that such an evaluation can only serve as a baseline evaluation measure due to the restriction that only properties that have already been used on the given data item, may resolve to true positives and hence, influence the evaluation result. I.e., additional properties that users might still consider useful or suitable for a given data item, cannot be evaluated as these naturally are not contained in the data set. Therefore, we strongly call for a user-based evaluation of such user-assisting measures to evaluate the usefulness of the proposed property recommendations from a user-perspective. This would also allow for evaluating to which extent property recommendations contribute to an increased number of properties per item—or put differently: whether recommendations foster user engagement by providing easy means for entering additional information. Such an evaluation should not only include the evaluation of pure recommendations, but should also integrate and evaluate user interface issues to validate the current design of how such recommendations are presented. This should obviously also be generalized to evaluating the usability of the user interface provided to edit data items and their characteristics.

Regarding the usability of the user interface provided to enter new information, value suggestions as well as source reference and qualifier suggestions for a given item and property have also been implemented (cf. the development plan for the Wikidata entity suggester[10] or the official website documenting the progress of the Wikidata Property Suggester[11]). The evaluation of such recommendations is out of scope of this particular analyses, however, seems to be interesting future work to us. The recommendation of values and according units have also been proposed and implemented in the Snoopy system [7], therefore a direct comparison and research on possible improvements is of great interest to us.

7. CONCLUSION

In this paper, we presented an evaluation of recommender algorithms aiming at assisting users in the process of entering information on the Wikidata platform. Therefore, we compare the Wikidata Entity Suggester with two property recommender systems stemming from related fields: the Predicate Suggestion approach by Abedjan and Naumann and the Snoopy approach by Zangerle and Gassler. We find that the current implementation of the Wikidata Entity Suggester works better than the other presented approaches. In the course of our analyses, we identify two key aspects which are essential for the quality of recommendations: incorporating classifying properties and making use of contextual information for ranking the property recommendation candidates. Combining the current Wikidata Entity Suggester approach with Snoopy's ranking strategy, which facilitates contextual information, significantly increases the performance of the current Wikidata recommender approach.

As for future work, we strive to perform a user-centric evaluation of these algorithms as well as the respective user interface to enable a comprehensive evaluation from a user's perspective. Moreover, we also plan to investigate the suitability of state-of-the-art recommendation algorithms not relying on AR based on e.g., matrix factorization.

[10] https://www.wikidata.org/wiki/Wikidata:Development_plan#Entity_suggester

[11] https://www.mediawiki.org/wiki/WikidataEntity\-Suggester/Progress

8. REFERENCES

[1] Z. Abedjan and F. Naumann. Improving rdf data through association rule mining. *Datenbank-Spektrum*, 13(2):111–120, 2013.

[2] Z. Abedjan and F. Naumann. *The Semantic Web: ESWC 2014 Satellite Events: ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, chapter Amending RDF Entities with New Facts, pages 131–143. Springer International Publishing, Cham, 2014.

[3] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(2):207–216, 1993.

[4] S. Burgstaller-Muehlbacher, A. Waagmeester, E. Mitraka, J. Turner, T. E. Putman, J. Leong, P. Pavlidis, L. Schriml, B. M. Good, and A. I. Su. Wikidata as a semantic framework for the gene wiki initiative. *bioRxiv*, 2015.

[5] P. Cremonesi, R. Turrin, E. Lentini, and M. Matteucci. An evaluation methodology for collaborative recommender systems. In *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on*, pages 224–231. IEEE, 2008.

[6] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandečić. *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, chapter Introducing Wikidata to the Linked Data Web, pages 50–65. Springer International Publishing, Cham, 2014.

[7] W. Gassler, E. Zangerle, and G. Specht. Guided Curation of Semistructured Data in Collaboratively-built Knowledge Bases. *Journal on Future Generation Computer Systems*, 31:111–119, 2014. impact factor 1.978.

[8] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM Sigmod Record*, volume 29, pages 1–12. ACM, 2000.

[9] D. Hernández, A. Hogan, and M. Krötzsch. Reifying rdf: What works well with wikidata? In *Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located with 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA*, pages 32–47, 2015.

[10] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[11] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[12] C. Müller-Birn, B. Karran, J. Lehmann, and M. Luczak-Rösch. Peer-production system or collaborative ontology engineering effort: What is wikidata? In *Proceedings of the 11th International Symposium on Open Collaboration*, OpenSym '15, pages 20:1–20:10, New York, NY, USA, 2015. ACM.

[13] A. Pfundner, T. Schönberg, J. Horn, R. D. Boyce, and M. Samwald. Utilizing the wikidata system to improve the quality of medical content in wikipedia in diverse languages: A pilot study. *Journal of medical Internet research*, 17(5), 2015.

[14] D. Sanchez, M. Vila, L. Cerda, and J. Serrano. Association rules applied to credit card fraud detection. *Expert Systems with Applications*, 36(2, Part 2):3630 – 3640, 2009.

[15] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[16] C. Schmitz, A. Hotho, R. Jäschke, and G. Stumme. *Data Science and Classification*, chapter Mining Association Rules in Folksonomies, pages 261–270. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[17] B. Sigurbjörnsson and R. Van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, pages 327–336. ACM, 2008.

[18] T. Steiner. Bots vs. wikipedians, anons vs. logged-ins (redux): A global study of edit activity on wikipedia and wikidata. In *Proceedings of The International Symposium on Open Collaboration*, OpenSym '14, pages 25:1–25:7, New York, NY, USA, 2014. ACM.

[19] T. H. Ta and C. Anutariya. *Semantic Technology: 4th Joint International Conference, JIST 2014, Chiang Mai, Thailand, November 9-11, 2014. Revised Selected Papers*, chapter A Model for Enriching Multilingual Wikipedias Using Infobox and Wikidata Property Alignment, pages 335–350. Springer International Publishing, Cham, 2015.

[20] J. J. Treinen and R. Thurimella. A framework for the application of association rule mining in large intrusion detection infrastructures. In *Recent Advances in Intrusion Detection*, pages 1–18. Springer, 2006.

[21] D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[22] D. Vrandečić. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pages 1063–1064, New York, NY, USA, 2012. ACM.

[23] D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, Sept. 2014.

[24] O. R. Zaiane. Building a recommender agent for e-learning systems. In *Computers in Education, 2002. Proceedings. International Conference on*, pages 55–59 vol.1, Dec 2002.

[25] E. Zangerle, W. Gassler, and G. Specht. Recommending Structure in Collaborative Semistructured Information Systems. In *RecSys '10: Proceedings of the third ACM conference on Recommender systems*, pages 141–145, Barcelona, Spain, 2010. ACM.