RESEARCH

Hybrid Semantic Recommender System for Chemical Compounds in Large-Scale Datasets

Marcia Barros[1,2*], Andre Moitinho[2] and Francisco M. Couto[1]

Abstract

The increasing number of Chemical Compounds is a challenge for the researchers to explore such datasets. In this work, we propose the use of Recommender Systems in the exploration of new Chemical Compounds of interest to scientific researchers. Our approach consists in a Hybrid recommender model suitable for implicit feedback datasets and focused in retrieving a ranked list according to the relevance of the items. The model integrates collaborative-filtering algorithms for implicit feedback (Alternating Least Squares (ALS) and Bayesian Personalized Ranking(BPR)) and a new content-based algorithm, based on the semantic similarity of the Chemical Compounds in the ChEBI ontology. The algorithms were assessed on an implicit dataset of Chemical Compounds, CheRM-20, with more than 16.000 items (Chemical Compounds). The Hybrid model was able to improve the results of the collaborative-filtering algorithms, with increases of more than 10 percentage points in most of the assessed evaluation metrics.

Keywords: Recommender System; Chemical Compound; Ontology; Semantic Similarity

Introduction

Chemical Entities or Chemical Compounds, defined as "physical entities of interest in chemistry including molecular entities, parts thereof, and chemical substance"[1], are growing in number and complexity, generating large datasets, difficult for the researchers to explore in a deep way. Recommender Systems (RS) may be the solution for this challenge, by finding new entities to explore, for example, by suggesting entities not yet studied by the researchers based on their past investigation projects. However, the recommendation of Chemical Compounds of interest has not been widely explored [2, 3]. One challenge to include RS in compound databases is the lack of available datasets with the preferences of the researchers about the Chemical Compounds for assessing the RS. For example, it is difficult to explicitly know if a specific researcher had interest in the study of a chemical or not. More recently, alternatives have emerged with the development of datasets consisting of data collected from implicit feedback [4, 5]. These datasets do not contain the explicit interests of the users, as other famous datasets, such as Movielens [6]. Instead, this information is extracted from their activities, mostly from the scientific literature, the main method for disseminating scientific work.

Datasets of explicit or implicit feedback require different recommender algorithms, especially because implicit feedback has some significant downgrades, such as the lack of negative feedback, and unbalanced ratio of positive vs unobserved ratings [7, 8]. When dealing with implicit feedback datasets, the solution involves applying learning to rank (LtR) approaches. LtR consists in, given a set of items, identify in which order they should be recommended [9].

In RS, the main approaches are Collaborative-Filtering (CF) and Content-Based (CB) [10]. CF uses the similarity between the ratings of the users, and CB uses the similarity between the features of the items. CF is divided in two methods, memory-based and model-based [11]. Memory-based methods deal with the recommendation problem by finding the most similar users based on the ratings of the items. If two users tend to rate the same items in the same way, they will probably like the items seen by each other. Model-based methods use machine learning and data mining for predicting the ratings or for assigning a score to each item, by filling the rating matrix blank

*Correspondence: mbarros@fc.ul.pt
[1]LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, 1749–016 Lisboa, Portugal
[2]CENTRA, Departamento de Física, Faculdade de Ciências, Universidade de Lisboa, 1749–016 Lisboa, Portugal
Full list of author information is available at the end of the article

spaces (unknown ratings). One of the most used methods is matrix factorization, since it leverages all row and column correlations in one shot to estimate the entire data matrix [12]. With model-based methods, it is more difficult to explain the recommendations.

CF approaches cannot deal with new items or new users in the system, i.e., items and users without ratings (cold start problem). CB does not suffer from cold start problem for new items, since this approach only needs the features of the new items to compare with the features of the items that the user already saw or liked. Thus, even if the new item does not have a single rating in the entire dataset, it stills may be recommended. However, CB needs a list of features for the items, which varies from field to field. To deal with CF and CB challenges, we can have Hybrid RS, which are the assembling of CF and CB. One of the most common forms of creating Hybrids is by a weighted technique, where the scores of the different algorithms are combined into a unique final score [13].

One of the challenges of CB approaches is related to which features to use for finding similar items. Some items have obvious features. For example, when our items are movies the features used to find similar items may be the genre, director, and authors. In other fields, the task of finding features for the items is not so obvious. Thus, one of the tools used by CB for this purpose are ontologies [14], which provide controlled vocabularies of terms and definitions to represent the entities of a specific field of study [15, 16]. Some examples of well-known ontologies are the Chemical Entities of Biological Interest (ChEBI) [17, 18], the Gene Ontology (GO) [19, 20], and the Disease Ontology (DO) [21, 22]. The ontologies may be used for finding similar items, for example, by finding the amount of semantics that two entities share.

The field of RS is vast and its approaches are applied to several domains, such as movies [23], books [24], and e-commerce [25]. In the Chemistry domain, RS have been generally used in studies related to drugs, for example for new drugs design [26], and for finding candidate drugs for diseases [27]. Most recently, [28] used RS approaches in the discovery of new antiviral drugs, extracting compounds from ChEMBL [29], a database of molecules with drug-like properties. Other applications of RS in Chemistry may be found in [2], which describes the use of CF methods for creating possibilities for new chemical compounds. [3] use RS techniques also for the discovery of new inorganic compounds. The authors used the features of chemical relevant compositions to predict if a certain composition is a good candidate to inorganic compound. If the system predicts a composition as being a new compound, it recommends this composition to further studies.

None of the previous studies reported the use of ontologies, opposed to the studies presented bellow, in which the CF approaches were enhanced by the introduction of ontologies. [30] created a RS for recommending English collections of books in a library. The authors developed PORE, a personal ontology Recommender System, which consists of a personal ontology for each user and then the application of a CF method. [31] also used an ontology for creating users' profiles for the domain of books. They calculated the similarity, not between the ratings of the users, but based on the interest scores derived from the ontology. [32] developed a Trust–Semantic Fusion approach, tested on movies and Yahoo! datasets. Their approach incorporates semantic knowledge to the items primary information, using knowledge from the ontologies.

[33] presented a solution for the top@k recommendations (list of size k with the most relevant items for a user predicted by the recommendation algorithm) specifically for implicit feedback data. The authors developed the Spank - semantic path-based ranking. They extracted path-based features of the items from DBpedia and used LtR algorithms to get the rank of the most relevant items. They tested the method on music and movies domains. [34] developed a new semantic similarity measure, the Inferential Ontology-based Semantic Similarity. The new measure improved the results of a user-based CF approach, based on tests on the tourism domain. Most recently, [35] developed a Hybrid RS tested on the movies domain. The method used Single Value Decomposition for dimensionality reduction for the item and user-based CF, and ontologies for item-based semantic similarity, improving the CF results. They do not deal with implicit data.

For datasets of implicit feedback there are two CF algorithms which have been particularly used, Alternating Least Squares (ALS) [36] and Bayesian Personalized Ranking (BPR) [7]. ALS is a latent factor algorithm that addresses the confidence of a user-item pair rating, which goal is to minimize the least squares of the rating matrix and the matrix resultant from the dot product of the user matrix and item matrix. ALS has the advantage of being easily parallelized. Some recent studies focused on speeding up the implementation of this algorithm [37, 38]. Other study, developed a recommender system for movies based on ALS using Apache Spark [39]. BPR is also a latent factor algorithm, but it is more appropriate for ranking a list of items. BPR does not just consider the unobserved user-item pairs as zeros, but also takes into consideration the preference of a user between an observed and an unobserved rating. Several studies have been using BPR in the recommendation of items from implicit feedback datasets. [40] presented a deep neural network model based on Stack Denoising Auto-Encoder

and BPR. [41] proposed a social distance-aware BPR model, for social networks recommendations. [42] presented a solution for the recommendation of restaurants, based on deep learning and BPR, for multi-source datasets of implicit feedback.

Here we present a new Hybrid semantic recommender model for recommending Chemical Compounds that uses semantic similarity and deals with implicit feedback data, of which a prototype has been presented in [43]. The system here presented is now capable of dealing with thousands of items and the results represent an improvement over top@k in several evaluation metrics. The Hybrid model has two modules, one CF and one CB. The CF module addresses the implicit feedback datasets by applying ALS or BPR, and the CB module explores the semantic similarity of the chemical compounds. The Hybrid model combines the outcomes of the CF and CB modules.

The main contributions of this work are:

- a recommender framework for recommending Chemical Compounds;
- a new CB semantic recommender algorithm based on ontologies;
- a new Hybrid recommender algorithm for datasets of implicit feedback;
- a dataset with the semantic similarity between more than 16.000 Chemical Compounds;
- a faster semantic similarity calculation for DiShIn library.

The framework developed for this work, as well as all the data, is available at `https://github.com/lasigeBioTM/ChemRecSys`.

Methods

Workflow of the proposed model

In this work we propose a Hybrid recommender model, featuring two two modules: CF and CB. Figure 1 shows the general workflow of the model.

Figure 1 hybrid algorithm. Workflow of the Hybrid recommender model.

The input data used in this model, better described in Experiments Section, has the format of <user,item,rating>. The unrated set represents the items we want to rank to provide the best recommendations in the first positions to a user. The rated set are the items the users already rated. Since we will split the data into train and test, lets call training set to the rated set and testing set to the unrated set. Both training and testing sets are the input for the CF and CB modules. Using CF algorithms for implicit feedback datasets, the CF module gives a score for each item in the test set. The CB module uses semantic similarity for providing a score for the items in the test set. In the last step, the scores from CF and CB modules are combined and sorted in descending order.

For the CF module, we selected two CF recommender algorithms for recommending data collected from implicit feedback, Alternating Least Squares (ALS) [36] and Bayesian Personalized Ranking (BPR)[7], both implemented in the Python Library Fast python collaborative filtering for implicit datasets (implicit)[44]. These algorithms and the implementation in the implicit library are suitable for the type of dataset we are using and they were already used with similar datasets, i.e., recommendation datasets of implicit feedback, specially for recommending music playlists [45, 46]. ALS and BPR are used separately in the CF module, with the goal of verify which combination of CF/CB achieve the best recommendations results. The CF module outputs a score, S_{CF}, for each test item.

To the CB module we developed a new algorithm, called ONTO, which is based on the semantic similarity between the items in the ChEBI ontology. This module assigns a score S_{CB} to each item in the test set, calculating the semantic similarity between each item in the train and the test sets, as shown in Figure 1. The semantic similarity allows measuring how close two entities are in a semantic base. When using ontologies, the semantic similarity may be measured, for example, by calculating the shortest path connecting the nodes of two entities. For calculating the similarity, we used DiShIn [47, 48], a tool for calculating semantic similarities between the entities represented by an ontology. DiShIn provides three similarity measures: Resnik [49], Lin [50], and Jiang and Conrath (JC) [51]. All the previous measures are based on the information content of the entities, given by the probability of the entity appears in the ontology, and in the shared information content, calculated from the common ancestors. Resnik and Lin are real similarity measures, whereas JC is a distance measures, posteriorly converted to similarity. Lin and JC has a range between zero and one, thus more easy to use. The ONTO algorithm is described in Algorithm 1.

```
Data: train = [I2, I3, I4], test=[I1]
Result: List of scores for each item in Test
test_scores = [ ];
for i in test do
    score_i = [ ];
    for b in train do
        score_i.append(sim(i,b))
    end
    test_score.append(score_i.mean())
end
```

Algorithm 1: ONTO algorithm.

ONTO receives as input two lists of items, train and test. The train data are the items we know the user already saw. The test data are the items we want to know if are suitable for recommending to a user. Thus, for each item in the test set, the ONTO algorithm finds the similarity to each item in the train set and calculates the mean of the similarities, as expressed by Equation 1.

$$S_{\text{CBI1}} = \frac{Sim_{1,2} + Sim_{1,3} + ... + Sim_{1,n}}{m} \quad (1)$$

In Equation 1, S_{CBI1} is the score for item 1, which is a test item, calculated through the ONTO algorithm, and $Sim_{1,2}$, $Sim_{1,3}$, $Sim_{1,n}$ are the semantic similarities between item 1 and items 2, 3, ..., n, respectively. 2, 3 and n are train items, and m is the number of train items.

Whereas the CF module uses all the ratings from the train set to train the model, CB module only takes into account the ratings of each user. ONTO algorithm does not use any real rating of the test items when calculating the score for each item in the test set, thus we do not have the problem of introducing bias in the results.

The final score for each item in the test set in the Hybrid model is the ensemble of the scores obtained from the CF algorithms, ALS or BPR, and the score obtained by the ONTO algorithm [13]. We used a weighted method, weighting the components heuristically according to two different metrics. Metric1 is represented in Equation 2 and it multiplies the scores from CF and CB approaches. Metric2 is represented in Equation 3 and it calculates the mean of the scores.

$$Metric1 = S_{\text{CFI1}} \times S_{\text{CBI1}} \quad (2)$$

$$Metric2 = \frac{S_{\text{CFI1}} + S_{\text{CBI1}}}{2} \quad (3)$$

S_{CFI1} is the score obtained for item 1, depending on the CF algorithm that we are using (ALS or BPR for our case study), and S_{CBI1} is the score for item 1 obtained with the CB algorithm. Metric2 (Equation 3) is a more standard approach, however, Metric1 (Equation 2) allows that items that are really outstanding in one of the algorithms are recommended. Our goal is to prove that by combining both modules, we can improve the results of each module separately.

Evaluation

One major challenge about RS is the evaluation of the algorithms. Most studies use offline evaluation, since it does not need interaction with real users, only requiring a dataset with the preferences of the users, for splitting into train and test sets. Depending on the goal of the algorithm, the type of evaluation will be different. There are algorithms whose goal is to predict the rating a user would give to an item, and other whose goal is to recommend a ranked list of items, i.e., the top@k items, where k is the size of the list. In the first case, these algorithms are evaluated for the predicted rating, using metrics such as Root Mean Squared Error (RMSE). RMSE measures the differences between the real rating of an item, and the rating predicted by a recommender algorithm, for all n items being analysed.

In the second case, when the algorithms return a ranked list of items, these may be evaluated for the number of relevant items recommended, for example through Precision (Equation 4), Recall (Equation 5) and F-Measure (Equation 6), and for the quality of the ranking, through Mean Reciprocal Rank (Equation 7) and Normalized Discounted Cumulative Gain (Equation 9).

$$Precision@k = \frac{relevant\_items@k}{k} \quad (4)$$

$$Recall@k = \frac{relevant\_items@k}{total\_relevant\_items} \quad (5)$$

$$F\_measure@k = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

$$MRR = \frac{1}{n\_users} \sum_{i=1}^{n\_users} \frac{1}{rank_i} \quad (7)$$

$$DCG = \sum_{i=1}^{n} \frac{relevance_i}{\log_2(i+1))} \quad (8)$$

$$nDCG = \frac{DCG}{iDCG} \quad (9)$$

Precision@k provides a measure of the relevant items that are recommended in the top@k list, recall@k the number of relevant items recommended in the top@k list, and f-measure provides an harmonic mean of precision and recall. The MRR evaluates in which position the first relevant item appears. The nDCG is an evaluation method which compares the ideal ranking of a test set (iDCG), with the ranking assigned by the recommendation algorithm (DCG - Equation 8) [52].

Experiments

The data used in this work is a subset of a recommendation dataset of Chemical Compounds, CheRM-20, with the format of <user,item,rating>, with 16.437 items, 2.193 users, and 117.020 ratings. CHeRM-20

was created in [5], using a methodology called LIBRETTI. This methodology allows the creation of standard recommendation datasets by using research literature for extracting implicit feedback for the researchers. This allows us to have a dataset with information about what chemical entities a researcher had interest in a period of time or in a corpus of articles. In CheRM-20, the users are authors from research articles and they have at least 20 items rated, the items are Chemical Compounds, which may be linked to ChEBI ontology, and the ratings are the number of articles the author wrote about the item [53]. The ratings were obtained through implicit feedback, thus the need to use algorithms suitable for this kind of data, such as ALS and BPR.

Table 1 shows the variation of algorithms evaluated in this study. For CF, we tested ALS and BPR, separately. We tested different latent factors, achieving the best results for this data with 150 factors. For CB, we tested the ONTO algorithm, using three different similarity measures: Lin, Resnik, and JC. The Hybrids were developed in combinations of the CF and CB approaches, using the two different metrics for calculating the final score of each item in the test set, Metric1 - Equations 2 and Metric2 - Equation 3.

Table 1 Variation of the algorithms evaluated.

CF	CB	Metric	Algorithm
ALS	-	-	ALS
BPR	-	-	BPR
-	ONTO_JC	-	ONTO_JC
-	ONTO_LIN	-	ONTO_LIN
-	ONTO_RESNIK	-	ONTO_RESNIK
ALS	ONTO_JC	Metric1	ALS_ONTO_JC_m1
ALS	ONTO_JC	Metric2	ALS_ONTO_JC_m2
ALS	ONTO_LIN	Metric1	ALS_ONTO_LIN_m1
ALS	ONTO_LIN	Metric2	ALS_ONTO_LIN_m2
ALS	ONTO_RESNIK	Metric1	ALS_ONTO_RESNIK_m1
ALS	ONTO_RESNIK	Metric2	ALS_ONTO_RESNIK_m2
BPR	ONTO_JC	Metric1	BPR_ONTO_JC_m1
BPR	ONTO_JC	Metric2	BPR_ONTO_JC_m2
BPR	ONTO_LIN	Metric1	BPR_ONTO_LIN_m1
BPR	ONTO_LIN	Metric2	BPR_ONTO_LIN_m2
BPR	ONTO_RESNIK	Metric1	BPR_ONTO_RESNIK_m1
BPR	ONTO_RESNIK	Metric2	BPR_ONTO_RESNIK_m2

We used offline methods based on preexisting datasets for evaluating the performance of the algorithms for the top@k, with k varying between 0 and 20, with steps of 1 [54]. From the vast range of metrics for evaluating recommender algorithms, we selected classification accuracy metrics and rank accuracy metrics, since they allow us to evaluate the algorithms for the relevant and irrelevant items recommended in a ranked list, and for the ability of an algorithm to recommend the items in the correct order. For this study, we use Precision, Recall (classification accuracy metrics), MRR and nDCG (rank accuracy metrics). All the selected metrics range between 0 and 1, with values closest to 1 better. For the segmentation of the dataset into training and testing sets, we used a cross-validation approach, by splitting users and items in 5 folds. Each iteration had 20% of the users and 20% of the items as test data, and 80% as train data. All the positive ratings in the test set are considered as relevant items for the user. We considered the unrated items as negative ratings, i.e., not relevant for the users. For the ONTO algorithm we also assessed how using the n most similar items affects the results, with n varying from 1, 5, 10, 15, 20, 25, 30 and all of the items.

The semantic similarity between the Chemical Compounds was calculated offline, using the DiShIn. Despite DiShIn robustness, the framework was not fit for a large number of items. Thus, we implemented a new functionality, Light DiShIn, which allowed to speedup the calculation of the the similarities and the feasibility of the ONTO algorithm. Light DiShIn was implemented based on Pandas [55], which is a python Framework for manipulating datasets, and the use of multiprocessing, introducing the use of multiple cores for processing the similarities. Table 2 and Figure 2 show the results of the speedup in latency (Equation 10 [56]) of Light DiShIn when compared with the original DiShIn. The number of similarities calculated (n similarities) are 1, 30, 60 and 180 and both systems calculated Resnik, Lin, and JC similarities metrics.

$$Speedup_{Latency} = \frac{Latency1}{Latency2} \tag{10}$$

Table 2 Evaluation of the speedup latency from original DishIn to Light DiShIn. The latency is measured in seconds and n similarities is the number of similarities calculated in each iteration of the test.

n similarities	Original DiShIn	Light DiShIn	Speed up
1	0.77	1.66	0.46
30	20.36	1.79	11.34
60	41.43	1.83	22.59
90	62.72	2.07	30.22
180	121.72	2.39	50.82

Figure 2 light dishin speedup Speedup of Light DiShIn with respect to the Original DiShIn.

According to the results, for calculating the similarity between two entities (n similarities = 1), the original DiShIn is faster. Though, when increasing the number of entities and the number of similarities for calculation, the Light DiShIn is much faster than the original DiShIn, whose calculation time seems to be exponential. In our tests, the speedup latency from original DishIn to Light DiShIn achieves values of 50 times

faster. For calculating the 131.538.810 similarities between the entities used for this work, we estimated that the original DiShIn would take 3.2 years. The similarities for 16.437 Chemical Compounds, 131.538.810 similarities, were calculated in less than a week and stored into a mySQL database for the measures Lin, Resnik and JC. This database is used by the ONTO algorithm for faster retrieving the semantic similarities of all items in the test and in the train sets. The introduction of Light DiShIn allows the viability of the execution of the ONTO algorithm, described in Algorithm 1.

Results and Discussion

We present the results of this study in Figure 3 for Precision and Recall, and in Figure 4 for MRR and nDCG, through the form of heat-maps, for all the algorithms in Table 1. The heat-maps show the results from top@1 to top@20, obtained using the 5 most similar items when calculating the scores for the ONTO algorithms, since these were the best results obtained. Following the heat-map, the more purple, the better the results. The Hybrids, both with ALS and BPR, achieved the best values for all the represented metrics. The best precision was obtained with ALS-ONTO-LIN-m2 (0.63 - top@1), improving the results of ALS by 7 percentage points. The best recall was obtained with ALS-ONTO-JC-m2 (0.55 - top@20), improving the results of ALS by 6 percentage points.

Figure 3 Precision and Recall. Precision and Recall results from top@1 to top@20, for ALS, BPR, ONTO and the Hybrids obtained using the 5 most similar items when calculating the scores for the ONTO algorithms.

Figure 4 MRR and nDCG. MRR and nDCG results from top@1 to top@20, for ALS, BPR, ONTO and the Hybrids obtained using the 5 most similar items when calculating the scores for the ONTO algorithms.

BPR had lower results than ALS for all the evaluated metrics, however, when BPR is combined with ONTO, the improvement is higher from BPR to BPR-ONTO than from ALS to ALS-ONTO. Precision had an improvement of 13 percentage points and recall had an improvement of 6 percentage points. From these results, we may conclude that the combination of ALS with ONTO achieves the highest results, but the hybrids with BPR undergo greater increases when compared to BPR alone. These results of precision and recall show that the Hybrid algorithms are including more relevant items in the list of recommendations.

Looking at the ranking quality metrics MRR and nDCG in Figure 4 the best results for MRR were obtained with ALS-ONTO-LIN-m2 (0.68 - top@20), with a growth of 7 percentage points from ALS to ALS-ONTO-LIN-m2. The best nDCG was registered with ALS-ONTO-JC-m2 (0.69 - top@20), more 6 percentage points than ALS. For BPR the increase was of 14 percentage points for MRR and 13 percentage points for nDCG. These results of MRR and nDCG indicate that the Hybrid algorithms are effective in rearranging the ranked list of recommendations.

Analysing Figures 3 and 4, the ONTO algorithms alone have the lowest results in all evaluation metrics. Nevertheless, they follow the trend of the other algorithms, and when measuring these metrics for the top@20, the results are similar. ONTO has the advantage of being a CB algorithm, therefore it does not have the problem of cold start for new items. ALS and BPR cannot be used if the item in the test set is not in the train set at least once (at least one author in the train set wrote about this Chemical Compound). However, ONTO algorithm requires the existence of all the entities in an ontology, in this case, the Chemical Compounds must be represented in ChEBI. ONTO-LIN and ONTO-RESNIK achieved almost the same results, however, the Hybrids created with the two metrics have quiet different results. The Hybrids with ALS created through Metric1 (Equation 2) achieved similar results for both ONTO-LIN and ONTO-RESNIK. For Metric2, the Hybrids with ONTO-LIN are better (Equation 3). This may be explained by the ranges of the scores. Whereas LIN have a range between 0 and 1, and ALS is also returning scores inferior to 1, the same is not true for ONTO-RESNIK, since Resnik similarity metric have an infinite upper limit. Thus, when using Metric2 for calculating the final score for an item, the scores from ONTO-RESNIK have a much greater influence in the mean of the scores than the ones from ALS (<1).

For BPR, we verified that the Hybrid with ONTO-RESNIK with Metric1 achieved similar results to the ones obtained with ONTO-LIN. With Metric2, the Hybrid with ONTO-RESNIK is better than with ONTO-LIN. Due to the particularity of BPR which always increments 1 to the scores, all scores for the items from this algorithm are higher than one. Between ALS and BPR, ALS achieved the best results. Since BPR is an algorithm for ranking, it was expected to obtain better results. We believe this is due the fact that the dataset has a large number of ratings equal to one, and many items have the same relevance (difficult to rank).

We will now see how the the number n of most similar items is also influencing the results of the ONTO algorithm, as well as the results for the Hybrids. Figure 5 shows the variation in the Precision@1,

Recall@20, MRR@20 and nDCG@20 with different n most similar items in the ONTO-RESNIK algorithm and for the Hybrids ALS-ONTO-RESNIK-m1, ALS-ONTO-RESNIK-m2, BPR-ONTO-RESNIK-m1 and BPR-ONTO-RESNIK-m2. ALS and BPR are also represented for a better visualization of the improvement of the Hybrids. The small variations of ALS and BPR along the y axis are due to the stochastic nature of the evaluation methods.

Figure 5 ONTO-RESNIK n variation. Variation of Precision@1, Recall@20, MRR@20 and nDCG@20 with different n most similar items in the ONTO-RESNIK algorithm.

Following Figure 5, the best results for ONTO-RESNIK in all the evaluation metrics are achieved using the 5 most similar items for calculating the scores of the items in the test set. Using a higher n, the quality metrics decrease for all the evaluation metrics. These results also affects the Hybrid algorithms, lowering the quality metrics with the increase of n. ALS-ONTO-RESNIK-m1 is the best for all evaluation metrics. Looking at the plots in Figure 5, we can notice a slightly descendent curve with the increase of the n most similar items. For example, the value for MRR@20 for ALS-ONTO-RESNIK-m2 is 0.6484 for n=5 and 0.6460 for n=10. This small difference may be because ALS has a much higher influence in the final score than ONTO-RESNIK. As previously noticed, ALS-ONTO-RESNIK-m2 suffers a decrease when compared with ALS. This is justified by the different ranges of the scores for each individual algorithm, visibly affecting ALS-ONTO-RESNIK-m2 by the variation of n. BPR follows the trend of the results for ALS, with the difference that BPR-ONTO-RESNIK-m2 generally achieved best results than BPR-ONTO-RESNIK-m1.

The results for the variation of the algorithms with the n most similar items for LIN and JC metrics are represented in Figures 6 and 7, respectively. The analysis of the plots suggest the same behaviour as the one for Resnik metric, i.e., the best results are achieved with n=5 and they degrade with the increase of n.

Figure 6 ONTO-LIN n variation. Variation of Precision@1, Recall@20, MRR@20 and nDCG@20 with different n most similar items in the ONTO-LIN algorithm.

The following example presented in Table 3 shows the influence of the ONTO-RESNIK algorithm in the order of the items in the ranked list of recommendations. The Table shows the top@20 recommended items with the algorithms ONTO-RESNIK, ALS, BPR, ALS-ONTO-RESNIK-m1, ALS-ONTO-RESNIK-m2, BPR-ONTO-RESNIK-m1 and BPR-ONTO-RESNIK-m2 for a user with ID 174228. This user has 4 relevant items (ChEBI ID/name: 85291 (N,1,2-trioleoyl-sn-glycero-3- phosphoethanolamine (1-)), 85292 (N-stearoyl-1,2-dioleoyl-sn-glycero-3- phosphoethanolamine (1-)), 137008 (N-acyl-1-[(1Z)-alkenyl]-sn-glycero-3- phosphoethanolamine (1-)) and 140452 (1-[(1Z)-octadecenyl]-2-oleoyl-sn-glycero-3-phosphate (2)) i.e., items in the test set with a rating higher than zero. The relevant items recommended by each algorithm are represented in gray cells.

Figure 7 ONTO-JC n variation. Variation of Precision@1, Recall@20, MRR@20 and nDCG@20 with different n most similar items in the ONTO-JC algorithm.

For the example presented in Table 3, the best algorithms were ALS, ALS-ONTO-RESNIK-m1, and BPR-ONTO-RESNIK-m2, following the trend of our general results presented in Figures 3, 4 and Figure 5. When combining ONTO-RESNIK with ALS with Metric1, the recommended items are the same, showing that for this case, ALS has a higher influence in the final results. Combining ONTO-RESNIK with ALS using Metric2 results in the recommendation of less relevant items in the first positions of the list. The Hybrid of ONTO-RESNIK and BPR using Metric1 or Metric2 improves the number of relevant items being recommended in the first positions for both BPR and ONTO-RESNIK. Based on these results, we may conclude that combining the ONTO algorithms with ALS or BPR, the most relevant items are rearranged for better positions in the Hybrids, improving the chances of recommending useful content for the users in the first positions of the recommendations. Thus, the results support our hypothesis that by using a CB algorithm based on the semantic similarity between the Chemical Compounds for creating Hybrids with CF algorithms, improves the recommendation of relevant items.

Considering that the size of the test set for this user was higher than 3000 items and the algorithms recommended 3 of the 4 relevant items in the first positions, one may say that RS are a solution for identifying Chemical Compounds of interest for scientific researches in large lists of these entities.

The development of Hybrid semantic RS based on our model may be easily applied to other areas, for example, for genes, phenotypes, and diseases, provided that exists an ontology for these items.

Conclusion

A major challenge in the discovery of new Chemical Compounds is the increasing number of entities being

Table 3 Influence of the ONTO-RESNIK algorithm in the top@20 list of recommendations for user 174228. This user has as relevant items the following ChEBI IDs: 85291, 85292, 137008 and 140452. The gray cells represent the relevant items recommended by each algorithm.

ONTO-RESNIK	ALS	BPR	ALS-ONTO-m1	ALS-ONTO-m2	BPR-ONTO-m1	BPR-ONTO-m2
85291	85292	23527	85292	85292	85292	85292
85292	85291	87818	85291	85291	85291	85291
85175	140452	72719	140452	85175	69120	140452
119	27847	6610	17697	119	140452	119
271436	175901	52347	5769	2904	137350	271436
2904	49668	72715	65495	271436	140243	6438
132187	87837	72754	27847	132187	132325	79079
79079	5769	69120	137411	79079	128770	132187
6438	17606	85292	49668	6438	69121	2904
140452	60453	140443	90983	140452	41214	69120
87764	87839	69340	132795	132725	82669	85175
132738	60747	132325	60999	132738	5635	137350
132725	76108	64499	30659	87764	63919	140243
65778	76097	140191	138802	78884	68249	128770
78884	60999	41214	138806	65778	69110	65778
76952	30659	91001	66917	141568	74912	69121
16108	31718	91000	37998	73275	140182	63919
77692	138802	133759	28850	138274	68236	69110
16125	138806	85291	66756	76952	130073	140182
31623	90983	67448	66755	16108	66394	130073

added to repositories. In this work, we presented a solution for this problem in the form of a recommender system. Our approach consists of a Hybrid recommender model for recommending ranked lists of Chemical Compounds. The Hybrid model has two modules based on the CF and CB approaches, respectively. In the CF module, we used ALS or BPR, specific algorithms for implicit feedback datasets. The CB module consists on a new algorithm called ONTO, based on the semantic similarity of the Chemical Compounds in ChEBI ontology. The hypothesis presented was that by combining the scores obtained by each module, we would improve the results of both modules separately. The Hybrids between ALS and ONTO were the ones with the best results for all the evaluation metrics, improving the results by more than 10 percentage points. The obtained results support our hypothesis, since the results for the Hybrids algorithms are higher when compared with the individual algorithms. Despite the fact that ALS and BPR are better than the ONTO versions of the CB approach, when combined, the ONTO algorithm rearranges the positions of the items, recommending more relevant items in the first positions of the rank. Thus, with this work, we contributed with a recommender framework for Chemical Compounds, a new CB semantic recommender algorithm based on ontologies, a new Hybrid recommender algorithm for datasets of implicit feedback, a dataset with the semantic similarity between more than 16.000 Chemical Compounds, and also a faster method for calculating the similarities between large numbers of entities. The presented methods may be extended to other fields of study, thereby, for future work we intent to assess the ONTO algorithm, as well as the Hybrids, with entities from other ontologies, such as GO and DO.

Availability of data and materials

The datasets supporting the conclusions of this article are available in the ChemRecSys GitHub repository, https://github.com/lasigeBioTM/ChemRecSys.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

MB and FC conceptualized the project. MB was responsible for the solution development. FC and AM supervised the project. All authors participated in the project discussion. All authors read and approved the final manuscript.

Acknowledgements

This work was supported by the Fundação para a Ciência e Tecnologia (FCT), under LASIGE Strategic Project UIDB/00408/2020, CENTRA Strategic Project UIDB/00099/2020, FCT funded project PTDC/CCI-BIO/28685/2017 and PhD Scholarship SFRH/BD/128840/2017.

Author details

[1]LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, 1749–016 Lisboa, Portugal. [2]CENTRA, Departamento de Física, Faculdade de Ciências, Universidade de Lisboa, 1749–016 Lisboa, Portugal.

References

1. ChEBI Entity "Chemical Entity". https://www.ebi.ac.uk/chebi/searchId.do?chebiId=CHEBI:24431 Accessed Accessed 24 Aug 2020
2. Ishihara, T., Koga, Y., Iwatsuki, Y., Hirayama, F.: Identification of potent orally active factor xa inhibitors based on conjugation strategy and application of predictable fragment recommender system. Bioorganic & medicinal chemistry **23**(2), 277–289 (2015)
3. Seko, A., Hayashi, H., Tanaka, I.: Compositional descriptor-based recommender system for the materials discovery. The Journal of chemical physics **148**(24), 241719 (2018)
4. Ortega, F., Bobadilla, J., Gutiérrez, A., Hurtado, R., Li, X.: Artificial intelligence scientific documentation dataset for recommender systems. IEEE Access **6**, 48543–48555 (2018)
5. Barros, M., Moitinho, A., Couto, F.M.: Using research literature to generate datasets of implicit feedback for recommending scientific items. IEEE Access **7**, 176668–176680 (2019)
6. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) **5**(4), 1–19 (2015)

7. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461 (2009). AUAI Press
8. Khawar, F., Zhang, N.L.: Conformative filtering for implicit feedback data. In: European Conference on Information Retrieval, pp. 164–178 (2019). Springer
9. Rendle, S., Balby Marinho, L., Nanopoulos, A., Schmidt-Thieme, L.: Learning optimal ranking with tensor factorization for tag recommendation. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 727–736 (2009). ACM
10. Ricci, F., Rokach, L., Shapira, B.: Recommender systems: introduction and challenges. In: Recommender Systems Handbook, pp. 1–34. Springer, Boston, MA (2015)
11. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in artificial intelligence **2009** (2009)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
13. Aggarwal, C.C.: Ensemble-based and hybrid recommender systems. In: Recommender Systems, pp. 199–224. Springer, Boston, MA (2016)
14. Tarus, J.K., Niu, Z., Mustafa, G.: Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. Artificial Intelligence Review **50**(1), 21–48 (2018)
15. Uschold, M., Gruninger, M.: Ontologies: Principles, methods and applications. The knowledge engineering review **11**(2), 93–136 (1996)
16. Barros, M., Couto, F.M.: Knowledge representation and management: a linked data perspective. Yearbook of medical informatics **25**(01), 178–183 (2016)
17. Chemical Entities of Biological Interest (ChEBI). https://www.ebi.ac.uk/chebi/ Accessed Accessed 24 Aug 2020
18. Hastings, J., Owen, G., Dekker, A., Ennis, M., Kale, N., Muthukrishnan, V., Turner, S., Swainston, N., Mendes, P., Steinbeck, C.: Chebi in 2016: Improved services and an expanding collection of metabolites. Nucleic acids research **44**(D1), 1214–1219 (2015)
19. Gene Ontology (GO). http://geneontology.org/ Accessed Accessed 24 Aug 2020
20. Consortium, G.O.: The gene ontology resource: 20 years and still going strong. Nucleic acids research **47**(D1), 330–338 (2018)
21. Disease Ontology (DO). http://disease-ontology.org/ Accessed Accessed 24 Aug 2020
22. Schriml, L.M., Mitraka, E., Munro, J., Tauber, B., Schor, M., Nickle, L., Felix, V., Jeng, L., Bearer, C., Lichenstein, R., *et al.*: Human disease ontology 2018 update: classification, content and workflow expansion. Nucleic acids research **47**(D1), 955–962 (2018)
23. Walek, B., Fojtik, V.: A hybrid recommender system for recommending relevant movies using an expert system. Expert Systems with Applications, 113452 (2020)
24. Tian, Y., Zheng, B., Wang, Y., Zhang, Y., Wu, Q.: College library personalized recommendation system based on hybrid recommendation algorithm. Procedia CIRP **83**, 490–494 (2019)
25. Shoja, B.M., Tabrizi, N.: Customer reviews analysis with deep neural networks for e-commerce recommender systems. IEEE Access **7**, 119121–119130 (2019)
26. Boström, J., Falk, N., Tyrchan, C.: Exploiting personalized information for reagent selection in drug design. Drug discovery today **16**(5-6), 181–187 (2011)
27. Hao, M., Bryant, S.H., Wang, Y.: A new chemoinformatics approach with improved strategies for effective predictions of potential drugs. Journal of cheminformatics **10**(1), 1–9 (2018)
28. Sosnina, E.A., Sosnin, S., Nikitina, A.A., Nazarov, I., Osolodkin, D.I., Fedorov, M.V.: Recommender systems in antiviral drug discovery. ACS omega (2020)
29. ChEMBL. https://www.ebi.ac.uk/chembl/ Accessed Accessed 24 Aug 2020
30. Liao, I.-E., Hsu, W.-C., Cheng, M.-S., Chen, L.-P.: A library recommender system based on a personal ontology model and collaborative filtering technique for english collections. The electronic library **28**(3), 386–400 (2010)
31. Sieg, A., Mobasher, B., Burke, R.: Improving the effectiveness of collaborative recommendation with ontology-based user profiles. In: Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 39–46 (2010). ACM
32. Shambour, Q., Lu, J.: A trust-semantic fusion-based recommendation approach for e-business applications. Decision Support Systems **54**(1), 768–780 (2012)
33. Ostuni, V.C., Di Noia, T., Di Sciascio, E., Mirizzi, R.: Top-n recommendations from implicit feedback leveraging linked open data. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp. 85–92 (2013). ACM
34. Al-Hassan, M., Lu, H., Lu, J.: A semantic enhanced hybrid recommendation approach: A case study of e-government tourism service recommendation system. Decision Support Systems **72**, 97–109 (2015)
35. Nilashi, M., Ibrahim, O., Bagherifard, K.: A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Expert Systems with Applications **92**, 507–520 (2018)
36. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 263–272 (2008). Ieee
37. Hao, T., Zheng, Z.: The implementation and optimization of matrix decomposition based collaborative filtering task on x86 platform. In: International Symposium on Benchmarking, Measuring and Optimization, pp. 110–115 (2019). Springer
38. Liang, Y., Zeng, S., Liang, Y., Chen, K.: Accelerating parallel als for collaborative filtering on hadoop. In: International Symposium on Benchmarking, Measuring and Optimization, pp. 123–137 (2019). Springer
39. Aljunid, M.F., Manjaiah, D.: Movie recommender system based on collaborative filtering using apache spark. In: Data Management, Analytics and Innovation, pp. 283–295. Springer, Boston, MA (2019)
40. Bi, Z., Zhou, S., Yang, X., Zhou, P., Wu, J.: An approach for item recommendation using deep neural network combined with the bayesian personalized ranking. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 151–165 (2019). Springer
41. Zhao, F., Shen, Y., Gui, X., Jin, H.: Sdbpr: Social distance-aware bayesian personalized ranking for recommendation. Future Generation Computer Systems **95**, 372–381 (2019)
42. Zhang, X., Luo, H., Chen, B., Guo, G.: Multi-view visual bayesian personalized ranking for restaurant recommendation. APPLIED INTELLIGENCE (2020)
43. Barros, M., Moitinho, A., Couto, F.M.: Hybrid semantic recommender system for chemical compounds. In: European Conference on Information Retrieval, pp. 94–101 (2020). Springer
44. Fast Python Collaborative Filtering for Implicit Datasets. https://implicit.readthedocs.io/en/latest/index.html Accessed Accessed 24 Aug 2020
45. Vall, A., Eghbal-Zadeh, H., Dorfer, M., Schedl, M., Widmer, G.: Music playlist continuation by learning from hand-curated examples and song features: Alleviating the cold-start problem for rare and out-of-set songs. In: Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, pp. 46–54 (2017)
46. Vall, A., Dorfer, M., Eghbal-Zadeh, H., Schedl, M., Burjorjee, K., Widmer, G.: Feature-combination hybrid recommender systems for automated music playlist continuation. User Modeling and User-Adapted Interaction **29**(2), 527–572 (2019)
47. DiShIn: Semantic Similarity Measures Using Disjunctive Shared Information. https://github.com/lasigeBioTM/DiShIn Accessed Accessed 24 Aug 2020
48. Couto, F., Lamurias, A.: Semantic similarity definition. Encyclopedia of bioinformatics and computational biology **1** (2019)
49. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007 (1995)
50. Lin, D., *et al.*: An information-theoretic definition of similarity. In: Icml, vol. 98, pp. 296–304 (1998). Citeseer
51. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008 (1997)
52. Schröder, G., Thiele, M., Lehner, W.: Setting goals and choosing metrics for recommender system evaluations. In: UCERSTI2 Workshop at the 5th ACM Conference on Recommender Systems, Chicago, USA,

vol. 23, p. 53 (2011)
53. CheRM: Chemical Compounds Recommender Matrix. https://github.com/lasigeBioTM/CheRM Accessed Accessed 24 Aug 2020
54. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender Systems Handbook, pp. 257–297. Springer, Boston, MA (2011)
55. Pandas Python Library. https://pandas.pydata.org/ Accessed Accessed 24 Aug 2020
56. Hennessy, J.L., Patterson, D.A.: Computer Architecture: a Quantitative Approach. Elsevier, Waltham, MA (2011)