Information Retrieval Journal (2022) 25:269–305
https://doi.org/10.1007/s10791-022-09408-9

Recommendations for item set completion: on the semantics of item co-occurrence with data sparsity, input size, and input modalities

I. Vagliano[1] · **L. Galke**[2,3] · **A. Scherp**[4]

Received: 13 May 2021 / Accepted: 6 March 2022 / Published online: 4 April 2022

Abstract
We address the problem of recommending relevant items to a user in order to "complete" a partial set of already-known items. We consider the two scenarios of citation and subject label recommendation, which resemble different semantics of item co-occurrence: relatedness for co-citations and diversity for subject labels. We assess the influence of the completeness of an already known partial item set on the recommender's performance. We also investigate data sparsity by imposing a pruning threshold on minimum item occurrence and the influence of using additional metadata. As models, we focus on different autoencoders, which are particularly suited for reconstructing missing items in a set. We extend autoencoders to exploit a multi-modal input of text and structured data. Our experiments on six real-world datasets show that supplying the partial item set as input is usually helpful when item co-occurrence resembles relatedness, while metadata are effective when co-occurrence implies diversity. The simple item co-occurrence model is a strong baseline for citation recommendation but can provide good results also for subject labels. Autoencoders have the capability to exploit additional metadata besides the partial item set as input, and achieve comparable or better performance. For the subject label recommendation task, the title is the most important attribute. Adding more input modalities sometimes even harms the results. In conclusion, it is crucial to consider the semantics of the item co-occurrence for the choice of an appropriate model and carefully decide which metadata to exploit.

Keywords Recommender systems · Autoencoders · Data sparsity · Cold start · Citation recommendation · Subject label recommendation

✉ I. Vagliano
i.vagliano@amsterdamumc.nl

L. Galke
Lukas.Galke@mpi.nl

A. Scherp
ansgar.scherp@uni-ulm.de

[1] Amsterdam University Medical Centers, Amsterdam, The Netherlands

[2] ZBW—Leibniz Information Centre for Economics, Kiel, Germany

[3] Present Address: Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands

[4] University of Ulm, Ulm, Germany

1 Introduction

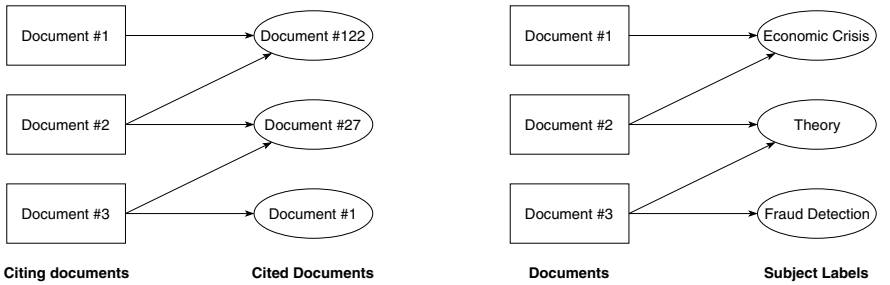
We address the problem of recommending items to a user in order to "complete" a partial set of items that the user already knows. Such a general recommendation task finds its applications in research paper recommendation (Beel et al., 2016; Raamkumar et al., 2017), citation recommendation (Caragea et al., 2013; Ebesu & Fang, 2017; Zhao et al., 2021), and multi-label classification (Tsoumakas & Katakis, 2007), which is known as subject indexing in the scientific digital-library community (ISO, 1996). In the latter, the task for professional indexers is to choose representative annotations from a domain-specific thesaurus in order to label scientific papers. The existing body of works is typically concerned with one-off recommendation, e.g., recommendations of scientific collaborators (Zhou et al., 2017), suggestion of which paper or news to read next (Beel et al., 2016; Hu et al., 2020; Cucchiarelli et al., 2019), or recommending an open sequence of items, e.g., music recommendations on Spotify (Bonnin & Jannach, 2014; Vagliano et al., 2018; Wang et al., 2018).

In contrast, we consider recommendation scenarios where there are items to be recommended to complete an existing, partial set of items. This set of items is complete at some point in time. For example, when writing a scientific article, the set of cited papers is accomplished at some point in time for the newly authored paper. This notion of completeness does not guarantee to have every possible related article covered with the citations, i.e., some papers may be missing due to various reasons, but the authored paper itself is finished and with it, the set of citations is concluded. In these set completion tasks, the number of possible outputs is as large as the power set of possible items, such that emitting crisp decisions, as usual in multi-label classification, is notoriously challenging (Tang et al., 2009; Nam et al., 2017, 2019). When we frame these problems as recommendation tasks, we rank the items rather than take a hard decision, and authors or subject indexers may benefit from recommendations until they consider the set to be complete. Below, we outline the considered recommendation scenarios in more detail.

1.1 Recommendation scenarios

We consider two recommendation scenarios (Fig. 1): Citation recommendation and subject label recommendation. Citations are known to resemble credit assignment (Wouters, 1999) and subject labels are selected by subject indexers such that all aspects of the paper are properly covered (ISO, 1996). As we discuss below, the two scenarios are chosen as they have different, complementing semantics of items occurring together in a set (co-occurring items). While in citation recommendation item co-occurrence implies similarity of the cited papers, in professional subject indexing, co-occurrence means dissimilarity of the annotated subjects. Thus, for citation recommendation, similar items will be sought, while labels that are different from those already assigned are naturally good recommendations for subject indexing. Despite the different underlying semantics, both scenarios can be modeled in a common framework. We take either the citations or the assigned labels as implicit feedback for the considered recommendation task.

Scenario 1: Item Co-occurrence as Similarity in Citation Recommendation When writing a new paper, the authors must reference other publications which are key in the respective field of study or relevant to the paper being written. Failing to do so can be rated negatively by reviewers in a peer-reviewing process. However, due to an increasing volume



(a) The citation recommendation scenario. **(b)** The subject indexing scenario.

Fig. 1 Exemplary bipartite graphs of citation relationships between documents (left) and documents annotated with subject labels (right) (Galke et al., 2018). The two recommendation tasks share a similar structure, but in the former the recommendation items are the cited documents, in the latter are the subject labels

of scientific literature, even some key papers are sometimes overlooked. We address this challenge by studying the problem of recommending publications to consider as citation candidates. Given that the authors have already selected some references for their scientific paper, we use these references as partial input set for recommendations of further documents that may be included as references. Initially, the set is very short since the completeness of the citation set depends on the stage of writing, while towards submission it is almost complete. If a document is at an early stage, where few other documents are cited, the chance to find a proper recommendation is high, but the information about which candidates to recommend is sparse as the set so far is rather small. In the middle stage of the writing process, more references have been included, but some more need to be found. In a late stage, most of the relevant documents have been added, making the set almost complete and the choices for further good recommendations small. In our experiments, we explicitly consider the influence of completeness of the partial input set on the recommendation performance.

Scenario 2: Item Co-occurrence as Dissimilarity in Subject Indexing Subject indexing is a common task in scientific libraries to make scientific documents accessible for search. New documents are manually annotated by qualified subject indexers with a set of subject labels, i. e., classes from a, typically domain-specific, thesaurus. Fully-automated multi-label classification approaches for subject indexing are promising (Nam et al., 2017), even when merely the metadata of the publications is used (Galke et al., 2017). Professional subject indexers, however, typically use the result of these approaches only as recommendations, in order to still guarantee a human-level quality. Similar to the citation task, the professional subject indexers start adding labels one by one to the publications. Following internationally standardized guidelines (ISO, 1996), the indexers ensure that the labels are covering well the often multiple scientific aspects, contributions, and methods of the papers. To this end, the indexers scan the paper's title, section headings, and partially read their content. Thus, properly indexing a scientific paper with subject labels is a difficult task that can be supported by a recommender system. The recommender takes the partial set of already assigned subject labels as input and generates recommendations for new labels that refer to aspects of the paper that are not yet covered. Following the same experimental approach as for the citation task, we explicitly evaluate the influence of the different levels of completeness of the already assigned subject labels on the recommendation

performance. Thus, we use different sizes of the partial input set of labels and measure the quality of the recommendations.

1.2 Research questions

In our experiments, we are interested in understanding the performance of recommending items to be added to an existing partial set under different conditions:

(i) Is there a difference in the recommender performance, when there are different underlying semantics of item co-occurrence in a set? We chose citation and subject label recommendation as our scenarios because of the difference in what it means when two items occur together (similarity or dissimilarity). These different semantics of item co-occurrence may have practical implications, e.g., regarding the choice of an appropriate model and which metadata attributes to exploit.

(ii) How does the completeness of the partial set of items influence the provided recommendations? Is it easier for the recommender to suggest good items at an early, mid, or late completeness stage for the set of items? We analyze the impact of the size of the partial set given as input by varying the number of dropped elements that need to be predicted for the output. The more items are dropped, the fewer items are available as input to the recommender (but more choices are available as correct items for the recommender). This question naturally renders the challenge of recommending items to a user to complete an existing, partial set of items, given different levels of completeness of this set, which represent different stages in the task of finding citations or subject labels.

(iii) How does pruning of rarely-cited documents, as well as documents which cite few other documents, affect the models' performance? Existing studies are often applied on datasets where rarely cited documents and documents that cite too few other works are removed (Beel et al., 2016). This is a problem as it is known that few papers have many more citations (the well-known power-law distribution Newman, 2005, or long-tail). This pruning step affects the number of considered items, and thus, the degree of sparsity. This prevents verifying how the recommender will behave in real-world settings. To systematically investigate how the pruning threshold affects the models' performance, we conduct experiments where the pruning threshold is a controlled variable.

(iv) What is the optimal use of additional bibliographic metadata of the documents? Using additional information in recommendation tasks has proven effective, and various kinds of information have been exploited, such as text (Chen et al., 2015), images (Zhang et al., 2016) and knowledge graphs (Zhang et al., 2016; Musto et al., 2017; Vagliano et al., 2017). However, an analysis of whether and how using bibliographic metadata is effective when considering tasks with different underlying semantics of item co-occurrence has not yet been conducted. We investigate the influence of different metadata attributes, such as authors, title, and venue, on the recommendation performance, while using the publication year to create a chronological train-test split.

1.3 Models and evaluation results

As recommender models, we focus on autoencoder architectures and compare them to strong baselines. Autoencoders have become very popular for recommendation tasks (Zhao et al., 2021; Pan et al., 2020; Liang et al., 2018; Steck, 2019), sometimes using side information (Bai & Ban, 2019; He et al., 2019; Chen & de Rijke, 2018; Liu et al., 2018). By learning to reconstruct their input, autoencoders are useful to recommend

items for the set completion task. Due to regularization, autoencoders can capture general patterns in the data rather than merely copying their input (Bengio et al., 2013). For our experiments, we consider four representative techniques to regularize autoencoders, namely undercomplete, denoising, variational, and adversarial autoencoders.

We extend the autoencoders to take multi-modal input such that they become hybrid recommenders that can exploit both the ratings in terms of the partial item set as well as the content in terms of additional metadata. These extended models can receive different metadata fields as input, such as the documents' title, authors, and venue. We compare autoencoders with three strong baselines for a fair evaluation and to address the question of whether neural architectures are making progress regarding the considered recommendation tasks (Dacrema et al., 2019). As baselines, we have selected a multi-layer perceptron (MLP), a classical recommendation model based on singular value decomposition (SVD), and a simple yet strong item co-occurrence baseline. MLP has been chosen because has previously shown good performance in subject label classification tasks (Galke et al., 2018; Mai et al., 2018).

We evaluate our models and baselines in both recommendation scenarios on three different datasets for each scenario. In total, we use six different datasets, which stem from five domains, namely medicine, computer science, economics, politics, and news. We pre-process the documents by splitting them into a training and testing set along the time axis, at a point t, i.e., our models are being trained on documents published before t and tested on the documents published after. This resembles the natural constraint that newly written publications can only cite already published works. We apply the same chronological split to subject labels to account for concept drift (Webb et al., 2018), i.e., changes in the joint distribution of documents and annotations over time. These settings are challenging as they correspond to a cold-start situation, more specifically, the well-known new user problem (Silva et al., 2019). We investigate each scenario regarding the influence of different proportions of completeness of the partial item set for the performance and different pruning of documents. Furthermore, we experiment with different combinations of metadata attributes as side information to the recommender. We run 4,590 experiments.

The key takeaway from our experiments is that supplying the partial item sets as input is usually most helpful when item co-occurrence resembles relatedness, while the content is effective when co-occurrence implies diversity. However, when item co-occurrence resembles relatedness, the content can be beneficial with short partial item sets and using subject labels as side information may be particularly effective in this case. Thus, when facing a new recommendation task, it is crucial to consider the semantics of item co-occurrence for the choice of an appropriate model and metadata attributes to exploit. Our results support this claim: In citation recommendation (co-occurring items are similar), the best performance is typically achieved without additional metadata, apart from PubMed with short partial item sets. In subject indexing (co-occurring items are diverse), using the content, e.g., with MLP or autoencoders conditioned on side information, is substantially more effective for predicting missing items than merely using the partial item set. We have confirmed these results on different degrees of sparsity (via controlled dataset pruning) and at different stages of the iterative set completion process.

1.4 Contributions and structure of the article

Below, we summarize our contributions. This paper presents significantly extended research based on our prior work (Galke et al., 2018). We added the research questions on the influence of the size of the partial input set, we used more datasets and models, which we also extended to exploit different metadata fields in a multi-modal fashion, and we provided a deeper analysis of how metadata affects the recommendation tasks.

1. We define a common representation for recommendation tasks in the context of set completion, where item co-occurrence resembles different semantics of the relation between items. This representation is based on the two real-world scenarios of citation recommendation, where item co-occurrence means relatedness, and subject label recommendation, where co-occurrence implies diversity.
2. We investigate the performance of the recommendation models along with four factors: (i) the semantics of item co-occurrence, (ii) the influence of the completeness of the partial set of items used as input, (iii) the pruning of infrequent items, and (iv) the optimal use of additional metadata available about the items. We extend different recommender architectures to incorporate metadata in addition to a partial set of items as input in a multi-modal fashion.
3. As experimental models, we focus on autoencoder architectures. They resemble a natural choice for the set completion task and are very popular for recommendation tasks. We compare four autoencoder architectures, with strong neural as well as non-neural baselines, to study the four factors (i) to (iv). This contributes to the pressing question of whether neural architectures are making a contribution to the state of the art in recommender systems (Dacrema et al., 2019).
4. We evaluate all autoencoder architectures and baselines in the two scenarios on six datasets from five different domains. The experimental data is split along the time axis, to resemble real-world settings.

In Sect. 2, we review previous work on recommender systems with a focus on autoencoder-based approaches and methods for citation and subject recommendation. In Sect. 3, we formally state the recommendation problem and introduce the extended multi-modal autoencoder models and baselines. We describe the experimental apparatus for the citation and subject recommendation experiments in Sect. 4. We present the results our experiments in Sect. 5 and discuss the results in Sect. 6, before concluding.

2 Related work

We review previous work on recommender systems with a focus on autoencoder-based approaches. Then, we discuss methods and systems specifically designed for citation and subject label recommendation.

2.1 Relation to collaborative filtering, content-based, and hybrid recommender systems

Recommender systems that recommend items to a user taking into account ratings that users with similar preferences gave to these items, i.e., operate only on the rating matrix $\mathbb{U} \times \mathbb{I}$, are typically labeled as *collaborative filtering* (Felfernig et al., 2013). On the other hand, approaches that take item content or ratings that users gave to items into account are referred to as *content-based* recommenders (Lops et al., 2011). *Hybrid* techniques combine these two approaches and consider both the content and user-item ratings. In hybrid recommenders, one can further distinguish between loose coupling and tight coupling (Wang et al., 2015). In loose coupling, collaborative filtering and content-based models recommend items separately and their output is subsequently combined. In tight coupling, a joint model operates on both input modalities. Our approaches behave as collaborative-filtering recommenders when only the partial set is given, while are content-based when only the metadata is given, and they behave as tightly coupled hybrid approaches when both the partial set and additional metadata are used.

2.2 Autoencoders for recommendation tasks

Autoencoders have recently gained wide popularity for recommendation tasks (Pan et al., 2020; Liang et al., 2018; Steck, 2019; Cao et al., 2017; Zhuang et al., 2017; Sedhain et al., 2015), oftentimes using side information (Bai & Ban, 2019; He et al., 2019; Chen & de Rijke, 2018; Liu et al., 2018; Wang et al., 2015; Majumdar & Jain, 2017; Li & She, 2017; Zhang et al., 2017; Strub et al., 2016; Li et al., 2015). Autoencoders are especially well-suited for tight-coupling collaborative filtering and content-based recommendations (Bai & Ban, 2019; Wang et al., 2015; Li & She, 2017). A common strategy is to combine auto-encoded item features with a latent factor model of the ratings (Bai & Ban, 2019; He et al., 2019; Li & She, 2017). Some works have also fused user/item side-information with the respective rows/columns of the rating matrix as input to the autoencoder (Chen & de Rijke, 2018; Majumdar & Jain, 2017). Majumdar and Jain (2017) investigated the pure cold start problem and partial cold start problems, in which they assume that 10% or 20% of the ratings are present. In contrast, we investigate additional stages of rating completeness. Regarding architectures, the variational autoencoder (VAE) (Kingma & Welling, 2014) has been most-often used for recommendation tasks (Liang et al., 2018; Bai & Ban, 2019; He et al., 2019; Chen & de Rijke, 2018; Li & She, 2017), although other works have used (stacked) denoising autoencoders (Liu et al., 2018; Wang et al., 2015; Li et al., 2015). Autoencoders have found widespread applications in recommender systems. The most prominent variants (Liang et al., 2018; Li & She, 2017) stood the test of a recent study on reproducibility in deep learning recommender systems (Dacrema et al., 2019), which has shown that all tested autoencoder variants (Collaborative VAE Li & She, 2017 and Mult-VAE Liang et al., 2018) could be reproduced.

Compared to previous works, we focus on using autoencoders for the new user problem (Silva et al., 2019) as modeled by a chronological split of the datasets along the time axis. In contrast, most of the existing literature assumes that all users are already present during training. Furthermore, we explicitly investigate the influence of different semantics of item co-occurrence on the recommendation performance. We also consider different completeness levels of the partial item set in our experiment for both our scenarios of citation and subject label recommendation, beginning from early to mid and late stages.

2.3 Research paper and citation recommendation

Research paper recommendation is a well-known and popular topic (Beel et al., 2016; Ali et al., 2021). Specifically for citation recommendation (Färber & Jatowt, 2020), one distinguishes between recommendations based on a partial set of references and recommendations based on the content of a manuscript (Huang et al., 2012). While the former strives to identify missing citations at the document level, the latter is suited for finding matching citations for a given statement, such as a sentence, during writing. Citation recommendation has recently focused on these context-sensitive applications, in which sentences are mapped to, preferably relevant, citations (Ma et al., 2020; Ebesu & Fang, 2017; Zhao et al., 2021; Beel et al., 2016; Huang et al., 2012). Instead, we revisit the reference set completion problem and we do not take the context of the citation into account, as the full text of a paper is rarely available in large-scale metadata sources (Mai et al., 2018). Co-citation analysis assumes that two papers are the more related to each other, the more they are co-cited (Small 1973). Following that idea, Caragea et al. relied on singular value decomposition as a more efficient and extendable approach for citation recommendation (Caragea et al., 2013). Other approaches made use of deep learning techniques for citation recommendation but focused on context-sensitive scenarios (Ebesu & Fang, 2017; Huang et al., 2015; Sakib et al., 2020; Zhang & Ma, 2020; Tao et al., 2020; Chen et al., 2020; Ali et al., 2020). Thus, we recognize the need for new methods for partial set completion problems that are not only based on item co-occurrence but also take additional metadata into account. While in our preliminary study (Galke et al., 2018), we have considered only title data, we now investigate the benefit of using more metadata, investigate the influence of different sizes of partial item sets, pruning of the items, and generalize the results on further datasets and models.

Further approaches from the areas of network analysis include node embeddings (Perozzi et al., 2014; Grover & Leskovec, 2016), link prediction with graph neural networks (Zhang & Chen, 2018), and dynamic graph representation learning (Kumar et al., 2019). However, these methods and also retrieval methods such as IRGAN (Wang et al., 2017) do not apply to our problem because they require all documents to be known in advance, while our methods need to apply to unseen data without further training. In graph representation learning, the former is known as transductive learning, while the latter is called inductive learning (Hamilton, 2020).

2.4 Subject label recommendation

Subject label recommendation is similar to tag recommendation: in both cases the goal is to suggest a descriptive label for some content. Boughareb et al. (2020) proposed an approach to recommend tags for scientific papers, which defines the relatedness between the tags attributed by users and the concepts extracted from the available sections of scientific papers based on statistical, structural, and semantic aspects . Sun et al. (2021) presented a hierarchical attention model for personalized tag recommendation. Lei et al. (2020) introduced a tag recommendation by text classification that uses the capsule network with dynamic routing for tag recommendation. The capsule network encodes the intrinsic spatial relationship between a part and a whole constituting viewpoint invariant knowledge that automatically generalizes to novel viewpoints. Zhou et al. (2020) proposed a novel hybrid method based on multi-modal content analysis that recommends keywords

to compose titles and tags of video uploaded to websites like YouTube, Yahoo Video, and Bing Video. They combined textual semantic analysis of original tags and recognition of video content with deep learning. Similarly, Sigurbjörnsson et al. proposed a tag recommender for Flickr to support the user in the photo annotation task (Sigurbjörnsson & van Zwol, 2008), whereas Posch et al. (2013) predicted hashtag categories on Twitter. While these works focused on tags for social media, we consider subjects from a standardized thesaurus for scientific documents.

Tag or label recommendation is also related to the problem of multi-label classification. Nam et al. (2017) proposed an encoder-decoder architecture based on sequence models as a special case of the generic set2set method (Vinyals et al., 2016). They extended their approach by an expectation-maximization model to supply context-dependent label permutations to the sequence mode (Nam et al., 2019). In our prior work, we analyzed titles vs full-text on scientific and news corpora for multi-label classification (Galke et al., 2017). We found that using only the title retains 90% of the accuracy obtained by using the full-text. Among many baselines, a wide MLP with a single hidden layer is preferable. Furthermore, we showed that using only the titles can even be better than using the full-text for multi-label classification tasks, when more training data is available (Mai et al., 2018). This motivates us to use additional bibliographic metadata in the two recommendation scenarios, rather than the full-text, which is often not available even in open access datasets (Mai et al., 2018).

2.5 Summary

Autoencoders have proven effective in recommendation tasks and have been widely used. To the best of our knowledge, no previous work has yet studied in detail the item recommendation at different stages of the partial set, i.e., with a varying number of items in input, and in the new user setting. Furthermore, we consider the differences in the semantics of item co-occurrence in the two recommendation scenarios of citation and subject label as well as different degrees of sparsity and the influence of using metadata.

3 Problem formalization and models

We first provide a formal problem statement for the considered recommendation tasks. The documents can be considered users in a traditional recommendation scenario, while the items are either cited documents or subject labels (Galke et al., 2018), respectively. Subsequently, we present the used autoencoders models. We focus on autoencoders, since they are specifically designed to reconstruct "corrupted" input.

3.1 Problem formalization

Traditionally, the recommendation problem is modeled as a matrix completion problem. The goal is the prediction of missing ratings in a $\mathbb{U} \times \mathbb{I}$ matrix, where $\mathbb{U}$ is the set of users and $\mathbb{I}$ is the set of items. As illustrated in Fig. 1, for citation recommendation (Scenario 1), the users are the newly authored papers and the recommended items are the papers that are recommended to be cited. While in subject label recommendation (Scenario 2), the users are the newly annotated papers, and the items are recommended subject labels. As done in previous work (McNee et al., 2002), we consider a matrix

Table 1 Notation table (Galke et al., 2018)

Symbol	Description
$\mathbb{D}$	Set of m documents
$\mathbb{I}$	Set of n items
$\boldsymbol{X} \in \{0,1\}^{m \times n}$	Sparse ratings matrix
$\boldsymbol{S} \in \mathbb{R}^{m \times d}$	Side information from document metadata
$\boldsymbol{x}, \boldsymbol{s}$	Row vectors of $\boldsymbol{X}$ or $\boldsymbol{S}$, respectively
$[\boldsymbol{x};\boldsymbol{s}]$	Concatenation of vectors $\boldsymbol{x}$ and $\boldsymbol{s}$
$\bowtie$	Natural join (on document identifiers)
$\boldsymbol{I}$	Identity matrix

$\mathbb{U} \times \mathbb{I}$, where the set of users is in our context the set of documents (i. e., the training corpus of documents). The rationale is that in Scenario 1, all authors for a given paper should receive the same recommendations of which papers to cite. Analogously for Scenario 2, a given paper should receive the same recommendations for adding subjects labels, independently of the current subject indexer in charge of annotating it.

Given a set of m documents, $\mathbb{D}$, and a set of n items, $\mathbb{I}$, the typical recommendation task is to model the spanned space, $\mathbb{D} \times \mathbb{I}$. We model the ratings as a sparse matrix $\boldsymbol{X} \in \{0,1\}^{m \times n}$, in which X_{jk} indicates implicit feedback from document j to item k. To simulate a real-world scenario, we split the documents $\mathbb{D}$ into m_{train} documents for training, $\mathbb{D}_{\text{train}}$, and m_{test} documents for evaluation, $\mathbb{D}_{\text{test}}$, such that $\mathbb{D}_{\text{train}} \cap \mathbb{D}_{\text{test}} = \emptyset$. More precisely, we conduct this split into training and test documents based on the publication year. All documents that were published before a certain year are used as training and the remaining documents as test data. This leads to an experimental setup that is close to a real-world application for citation recommendation and subject label recommendation. More details are provided in Sect. 4.2. All models are supplied with the users' ratings $\boldsymbol{X}_{\text{train}} = \mathbb{D}_{\text{train}} \bowtie \boldsymbol{X}$ along with the side information $\boldsymbol{S}_{\text{train}} = \mathbb{D}_{\text{train}} \bowtie \boldsymbol{S}$ for training. As side information $\boldsymbol{S}$, we use the documents' various metadata fields such as title, authors, and venue. The test set $\boldsymbol{X}_{\text{test}}, \boldsymbol{S}_{\text{test}}$ is obtained analogously. A summary of our notation can be found in Table 1.

For evaluation, we remove randomly selected items in $\boldsymbol{X}_{\text{test}}$ by setting a fraction of the non-zero entries in each row to zero. We denote the hereby created test set by $\tilde{\boldsymbol{X}}_{\text{test}}$. The model ought to predict values $\boldsymbol{X}_{\text{pred}} \in \mathbb{R}^{m_{\text{test}} \times n}$, given the test set, $\tilde{\boldsymbol{X}}_{\text{test}}$, along with the title information, $\boldsymbol{S}_{\text{test}}$. Finally, we compare the predicted scores, $\boldsymbol{X}_{\text{pred}}$, with the true ratings, $\boldsymbol{X}_{\text{test}}$, via ranking metrics. The goal is that those items, that were omitted in $\tilde{\boldsymbol{X}}_{\text{test}}$ are highly ranked in $\boldsymbol{X}_{\text{pred}}$.

In both scenarios, i.e., citation recommendation and subject label recommendation, we regard documents and items as a bipartite graph, as depicted in Fig. 1. Considering citations, this point of view may be counter-intuitive since a scientific document is typically both a citing paper and a cited paper. Still, the out-degree of typical citation datasets is so high that there are an order of magnitude more papers in the set of cited papers than in the set of citing papers of the bipartite graph. For instance, the PubMed citation dataset has 224,092 documents that cite 2,896,764 distinct other documents. Therefore, it is reasonable to distinguish the documents based on their role in the citation relationship, i. e., citing versus cited paper.

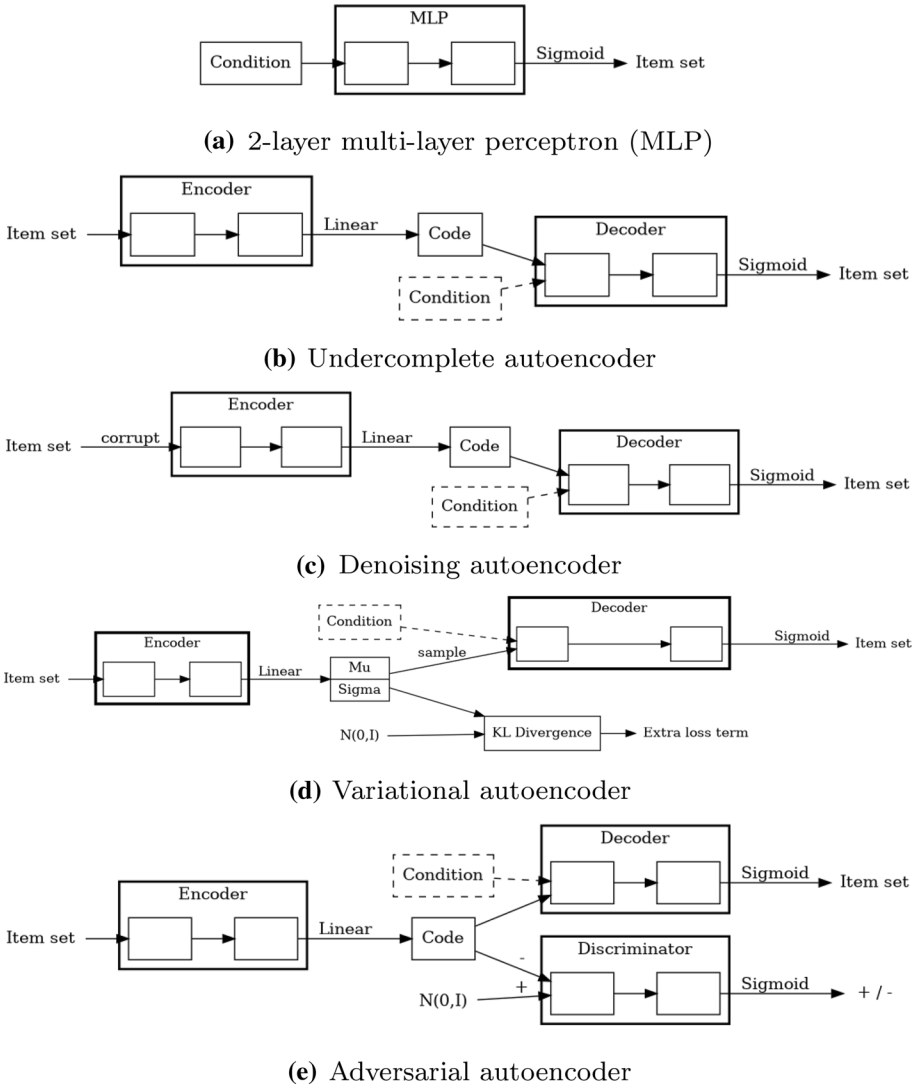


Fig. 2 Considered autoencoder architectures based on a 2-layer MLP as building block during training: **a** MLP baseline and its use as encoder and decoder in the autoencoders for item-based recommendations **b**–**e**. Each encoder/decoder/discrimination block is an MLP module. When not labeled differently, the activation function is ReLU followed by dropout. The condition is the supplied metadata. Only denoising autoencoder **c** corrupts the input item set during training, while the other architectures train the code on the full item set

3.2 Autoencoder models for itemset reconstruction

Autoencoders are particularly suited for our task as they aim to reconstruct corrupted input. Here, corruption means missing items. Thus, we focus on autoencoder architectures and compare them with strong baselines (introduced in Sect. 4.3). Below, we introduce the multi-layer perceptron as a building block for the autoencoders, and we show how metadata

can be incorporated in the former as well as in undercomplete, denoising, variational, and adversarial autoencoders.

Multi-Layer Perceptron

A multi-layer perceptron (MLP) is a fully-connected feed-forward neural network with one or multiple hidden layers (Fig. 2a). The output is computed by consecutive applications of $\mathbf{h}^{(i)} = \sigma(\mathbf{h}^{(i-1)} \cdot \mathbb{w}^{(i)} + \mathbf{b}^{(i)})$, with σ being a nonlinear activation function. In the description of the following models, we abbreviate a two-hidden-layer perceptron module by MLP-2.

Undercomplete Autoencoders An autoencoder (AE) has two main components: the encoder `enc` and the decoder `dec` (Fig. 2b). The encoder transforms the input $\mathbf{x}$ into a hidden representation, the code $\mathbf{z} = \text{enc}(\mathbf{x})$. The decoder aims to reconstruct the input from the code $\mathbf{r} = \text{dec}(\mathbf{z})$. The two components are jointly trained to minimize the binary cross-entropy, $\text{BCE}(\mathbf{x}, \mathbf{r})$. To avoid learning to merely copy the input $\mathbf{x}$ to the output $\mathbf{r}$, autoencoders need to be regularized. The most common way to regularize autoencoders is by imposing a lower dimensionality on the code (undercomplete autoencoder). In short, autoencoders are trained to capture the most important explanatory factors of variation for reconstruction (Bengio et al., 2013). For both the encoder and the decoder we chose an MLP-2 module, such that the model function becomes $\mathbf{r} = MLP-2_{dec}(MLP-2_{enc}(\mathbf{x}))$.

Denoising Autoencoders A denoising autoencoder (DAE) is an autoencoder that receives corrupted data as input and aims to predict original, uncorrupted data (Vincent et al., 2008) (Fig. 2c). During the training, the initial input $\boldsymbol{x}$ is corrupted into $\tilde{\mathbf{x}}$ through stochastic mapping $\tilde{\mathbf{x}} \sim q_D(\tilde{\mathbf{x}}|\mathbf{x})$, e.g., randomly forcing a fraction of entries to zero (white noise). The corrupted input $\tilde{\mathbf{x}}$ is then mapped to a hidden representation (the code) $\mathbf{z} = \text{enc}(\tilde{\mathbf{x}})$ in the same way of the standard autoencoder, and from the hidden representation the model reconstructs $\mathbf{r} = \text{dec}(\mathbf{z})$. Both the encoder and the decoder rely on a MLP-2 module, such that the model function becomes $\mathbf{r} = MLP-2_{dec}(MLP-2_{enc}(\tilde{\mathbf{x}}))$. The loss function is again the binary cross-entropy.

Variational Autoencoders A variational autoencoder (VAE) is a generative model whose posterior is approximated by a neural network, forming an autoencoder architecture (Kingma & Welling, 2014; Rezende et al., 2014) (Fig. 2d). Variational autoencoders use a variational approach for latent representation learning, in which the underlying data-generating distribution $p_{\text{data}}(\mathbf{X}|\mathbf{Z})$ is assumed to be a mixture of latent variables $\mathbf{z}$. The encoder $q_\theta(\mathbf{Z}|\mathbf{X})$ learns to infer these latent variable, while the decoder $p_\theta(\mathbf{X}|\mathbf{Z})$ learns to generate samples from these latent variables, where ϕ and θ are the parameters of the encoder and decoder, respectively. The goal is that $p_\theta(\mathbf{x}|\mathbf{z})$ approximates p_{data} after successful training. A crucial component of the VAE is the reparametrization trick that facilitates backpropagation through random operations (Kingma & Welling, 2014). Given a selected prior distribution $p_{\text{prior}}(\mathbf{Z})$ on the code $\mathbf{Z}$, a deterministic encoder learns to predict the parameters of this distribution. The prior distribution $p_{\text{prior}}(\mathbf{Z})$ of a VAE is typically Gaussian (Kingma & Welling, 2014). The deterministic output of the encoder $\text{enc}(\mathbf{x})$ is first split into two halfs: one for the mean, μ, and one for the standard deviations, σ. Then, we independently sample $\epsilon \sim \mathcal{N}(0, 1)$ and compose the code as $\mathbf{z} = \mu + \sigma\epsilon$ to be fed into the deterministic decoder. To encourage a normal distribution on the code, the Kullback-Leibler (KL) divergence with respect to $\mathcal{N}(0, 1)$ is added as an extra loss term. Intuitively, the encoder learns how much noise to inject at the code level. We use an MLP-2 in both the encoder and the decoder and optimize reconstruction loss (via binary cross-entropy) along with the KL divergence term.

Adversarial Autoencoders Adversarial autoencoders (AAE) (Galke et al., 2018; Makhzani et al., 2015) combine generative adversarial networks (Goodfellow et al., 2014) with autoencoders (Fig. 2e). The autoencoder component reconstructs the sparse item

Table 2 Availability and occurrence of metadata in the datasets considered for the two recommendation tasks

Metadata field	PubMed	DBLP	ACM	EconBiz	IREON	Reuters
Title	100%	100%	100%	100%	100%	100%
Author	100%	100%	93%	98%	83%	–
Abstract	–	89%	4%	–	–	–
Venue/Journal title	100%	83%	100%	–	–	–
Subject labels	77%	–	–	72%	100%	100%
Item set	100%	89%	81%	72%	100%	100%

Subject labels and item set occurrences are the same for the subject label datasets as the subject labels are the items to recommend (but can also be used as additional metadata for the citation tasks)

vectors, while the discriminator distinguishes between the generated codes and samples from a selected prior distribution. Hence, the distribution of the latent code is shaped to match the prior distribution. The latent representations learned by distinguishing the code from a smooth prior lead to a model that is more robust to sparse input vectors than undercomplete autoencoders because smoothness is the main criterion for good representations that disentangle the explanatory factors of variation (Bengio et al., 2013). Formally, we first compute $\mathbf{z} = \text{MLP-2}_{\text{enc}}(\mathbf{x})$ and $\mathbf{r} = \text{MLP-2}_{\text{dec}}(\mathbf{z})$ and then update the parameters of the encoder and the decoder with respect to the binary cross-entropy, $\text{BCE}(\mathbf{x}, \mathbf{r})$. Hence, in the regularization phase, we draw samples $\tilde{\mathbf{z}} \sim \mathcal{N}(0, \boldsymbol{I})$ from independent Gaussian distributions matching the size of $\mathbf{z}$. The parameters of the discriminator $\text{MLP-2}_{\text{disc}}$ are then updated, to minimize $\log \text{MLP-2}_{\text{disc}}(\tilde{\mathbf{z}}) + \log(1 - \text{MLP-2}_{\text{disc}}(\mathbf{z}))$ (Goodfellow et al., 2014). Finally, the parameters of the encoder are updated to maximize $\log \text{MLP-2}_{\text{disc}}(\mathbf{z})$, such that the encoder is trained to fool the discriminator. Thus, the encoder is jointly optimized for matching the prior distribution and reconstructing the input (Makhzani et al., 2015). At prediction time, we perform one reconstruction step by applying one encoding and one decoding step.

Conditioning on Additional Metadata as Side Information An advantage of autoencoder-based recommender systems is that they enable conditioning on side information. This conditioning may be performed with different strategies, e.g., by supplying the side information as additional input (Barbieri et al., 2017). In this work, we chose to impose the condition $\mathbf{s}$ on the code level of the autoencoders. The rationale for this strategy is that the side information may help the decoder to reconstruct the item set. Such strategy is similar to the supervised case of the original work on the adversarial autoencoder that used classes as condition to reconstruct images (Makhzani et al., 2015). It has the advantage that we can apply it in the same way to all autoencoder variants without further adaptions specific to an individual variant (such as disabling DAE's noise or modifying the VAE's KL divergence objective). This way, the encoder still operates solely on the (partial) item set while the decoder is conditioned on side information to reconstruct the original item set: $\mathbf{r} = \text{dec}(\text{enc}(\mathbf{x})||\mathbf{s})$, where $\cdot||\cdot$ denotes concatenation. Furthermore, the training objective remains optimizing for the reconstruction of the item set rather than reconstruction of the side information. In practice, we embed the documents' titles into a lower-dimensional space by using pre-trained word embeddings such as Word2Vec (Mikolov et al., 2013). More precisely, we employ a TF-IDF weighted bag of embedded words representation which has proven to be useful for information retrieval (Galke et al., 2017). We use the

same strategy for journal names as they often contain indicative words. For authors, we learn a categorical embedding from scratch: we optimize a randomly initialized embedding vector for each author during training. In our scenarios, the side information is composed of the documents' title, journal name, and authors. We consider three cases for our experiments: (1) no conditioning, (2) conditioning on the title, and (3) conditioning on all metadata (see Table 2). When multiple conditions are used, we combine them, again, via concatenation: $\mathbf{s} = \mathbf{s}_{\text{title}}||\mathbf{s}_{\text{author}}||\mathbf{s}_{\text{journal}}$. In our experiments, we evaluate those three cases with all the autoencoder variants described above.

4 Experimental apparatus

In the following, we describe the datasets, our experimental procedure, the baselines used, and the hyperparameter optimization.

4.1 Datasets

We consider six datasets for our experiments. Three datasets collect scientific publications in the domains of medicine and computer science for the citation recommendation task. Further, three datasets are in the domains of economics, politics, and news, and are used for the subject label recommendation task. Table 2 illustrates the metadata available in each dataset. For each dataset, we estimate the power-law coefficient and mutual information. The former allows us to assess whether and how the distribution of documents' citations and labels' assignments is skewed. The latter indicates how informative already assigned items are for other items. We estimate the power law coefficient α via maximum likelihood (Newman, 2005): $\alpha = 1 + n\left(\sum_{u \in V} \ln \frac{\deg_u}{\deg_{\min}}\right)^{-1}$, where $\deg_{\min}$ is equal to 1. We compute the mutual information, i.e., the KL Divergence of the joint distribution with respect the product of marginals: $\text{MI}(X, Y) = D_{KL}(p(x, y)||p(x)p(y))$ (Cover & Thomas, 2006). The distribution $p(x, y)$ models the number of documents in which two items occur together, while the marginals $p(x)$ and $p(y)$ are estimated based on the number of documents that have the item x. As $p(x)$ equals $p(y)$ in our case, the normalized mutual information is $\frac{MI(X,X)}{H(X)}$, with $H(X)$ being the entropy of X for normalization.

4.1.1 Datasets for citation recommendation

PubMed Citation Dataset The CITREC[1] PubMed citation dataset (Gipp et al., 2015) consists of 7,546,982 citations. The dataset comprises 224, 092 distinct citing documents published between 1928 and 2011 and 2, 896, 764 distinct cited documents. Each document has an identifier, article title, title of the journal where is published, list of authors, Medical Subject Headings (MeSH)[2] labels, and the publication year. The documents are cited between 1 and 3, 247 times with a median of 1 and a mean of 2.61 (SD: 6.71). The citing documents hold on average 33.68 (standard deviation, SD: 27.49) citations to other documents (minimum: 1, maximum: 2, 242) with a median of 29. This dataset, like all the

[1] https://www.isg.uni-konstanz.de/projects/citrec/.
[2] https://www.nlm.nih.gov/mesh/.

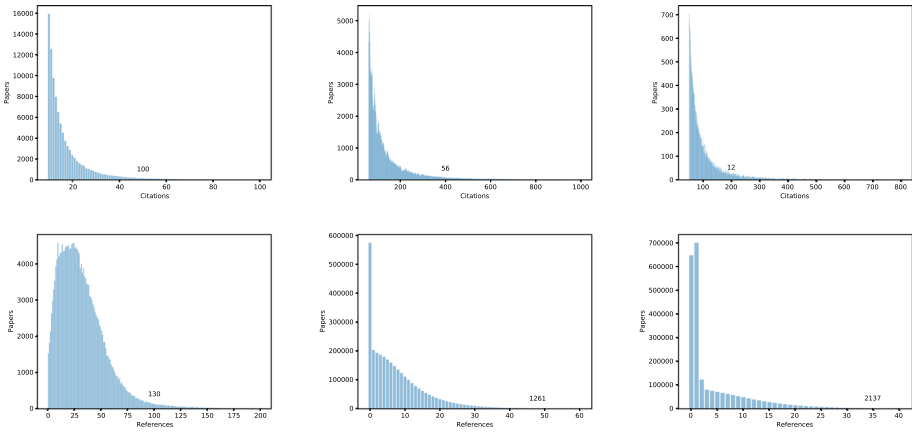


Fig. 3 Documents by citations for PubMed (top-left), DBLP (top-center) and ACM (top-right) datasets, and by the number of cited documents for PubMed (bottom-left), DBLP (bottom-center) and ACM (bottom-right)

others but Reuters, seems to follow a power-law distribution, which is typical for citation networks (Fig. 3). The α coefficient for the PubMed dataset is 1.47 for citations and 1.30 for the number of cited documents. The normalized mutual information for PubMed is 0.5996.

DLBP Citation Dataset The DLBP Citation Network[3] (Tang et al., 2008) includes 25, 166, 994 citations. The dataset comprises 3, 079, 007 distinct citing documents published between 1936 and 2018 and 1, 985, 921 distinct cited documents. Each document has an identifier, title, publication venue, authors, number of citations, publication date, and optionally the abstract. The documents are cited between 1 and 16, 229 times with a median of 4 and a mean of 12.67 (SD: 56.17). The citing documents hold on average 8.17 (SD: 9.71) citations to other documents (minimum: 0, maximum: 1, 532) with a median of 6. The α coefficient is 1.58 for citations and 1.28 for the number of cited documents. The normalized mutual information is 0.5407.

ACM Citation Dataset The ACM Citation Network[3] (Tang et al., 2008) contains 11, 344, 141 citations. The dataset comprises 2, 385, 066 distinct citing documents published between 1936 and 2016 and 2, 631, 128 distinct cited documents. Each document is characterized by its identifier, title, publication venue, authors, publication date, and optionally the abstract. The documents are cited between 0 and 810 times with a median of 1 and a mean of 4.76 (SD: 7.74). The citing documents hold on average 4.31 (SD: 580.96) citations to other documents (minimum: 1, maximum: 938, 039) with a median of 1. The α coefficient is 1.53 for citations and 1.32 for the number of cited documents, while the normalized mutual information is 0.5282.

4.1.2 Datasets for subject labels recommendation

EconBiz Subject Labels The EconBiz dataset,[4] provided by ZBW — Leibniz Information Centre for Economics, consists of 61, 619 documents with label annotations from

[3] https://aminer.org/citation.

[4] https://www.kaggle.com/hsrobo/titlebased-semantic-subject-indexing.

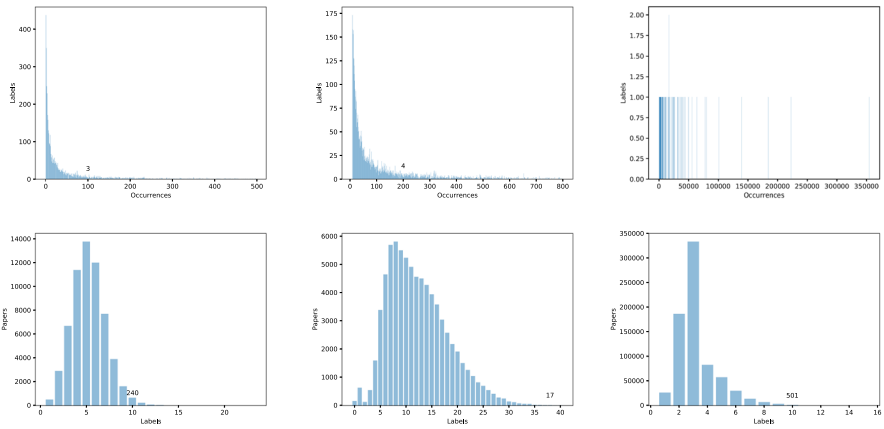


Fig. 4 Subject labels by the number of times they have been assigned to documents (occurrence) for the Economics (top-left), IREON (top-center), and Reuters (top-right) datasets, and documents by the number of labels they have been assigned for EconBiz (bottom-left), IREON (bottom-center), and Reuters (bottom-right)

professional subject indexers (Galke et al., 2017; Große-Bölting & Scherp, 2015). The 4, 669 assigned labels are a subset of the controlled vocabulary *Thesaurus for Economics*,[5] a professional thesaurus for economics. Every document has an identifier, title, authors, language label(s), subject labels, and publication year, as well as optionally the publisher, publication country, and series. The number of documents to which a subject label is assigned ranges between 1 and 13, 925 with a mean of 69 (SD: 316) and a median of 14. The label annotations of a document range between 1 and 23 with a mean of 5.24 (SD: 1.83) and a median of 5 labels. As for citations, the subject label datasets follow a power-law distribution, except Reuters (see Fig. 4). The α coefficient for the EconBiz dataset is 1.96 for the label occurrences and 1.19 for the number of assigned labels. The normalized mutual information is 0.2970.

IREON Subject Labels The political sciences dataset IREON has 76, 359 documents provided by the German Information Network for International Relations and Area Studies.[6] Each document holds an identifier, title, authors, language label, subject labels, and publication year. The 10, 440 subject labels assigned to the papers are taken from the thesaurus for International Relations and Area Studies.[7] The number of documents to which a label is assigned ranges between 1 and 13, 895 with a mean of 90.68 (SD: 338.63) and a median of 13. The label annotations of a document range between 0 and 70 with a mean of 12.40 (SD: 5.91) and a median of 11. The α coefficient is 1.94 for the label occurrences and 1.21 for the number of assigned labels, while the normalized mutual information is 0.1977.

Reuters Subject Labels The Reuters RCV1-v2 dataset contains 744, 693 news articles and a thesaurus providing a set of 104 labels (Lewis et al., 2004). Every document includes an identifier, title, subject labels, and publication date. The number of documents to which a label is assigned ranges between 5 and 354, 437 with mean 23, 217.35 (SD: 47, 313.01)

[5] http://zbw.eu/stw/version/latest/about.
[6] http://www.fiv-iblk.de/eindex.htm.
[7] http://www.fiv-iblk.de/information/information_thesaurus.htm.

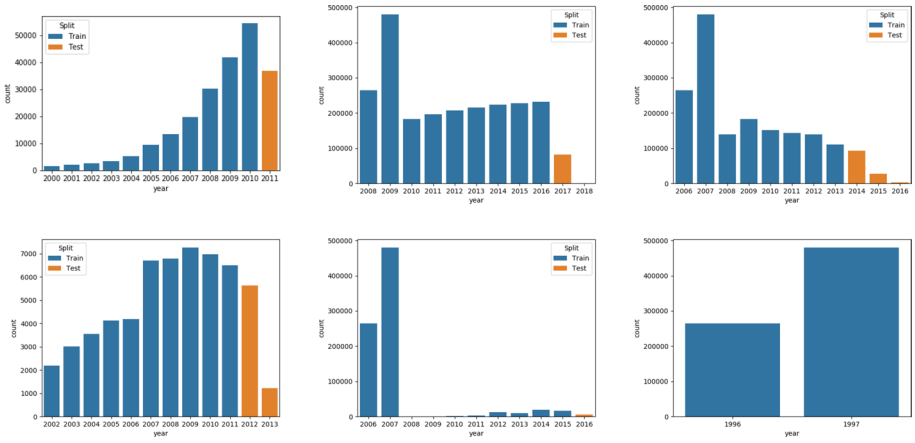


Fig. 5 Count of documents by publication year with the split in training and test set for PubMed (top left), DBLP (top center), ACM (top right) EconBiz (bottom left), IREON (bottom center), and Reuters (bottom right) datasets. For Reuters, the training and test sets are randomly generated

and median 7, 315. The label annotations of a document range between 1 and 17 with a mean of 3.24 (SD: 1.42) and a median of 3 labels. Its label occurrence does not follow a power-law distribution (Fig. 4). The α coefficient is 148.15 for label occurrence and 1.14 for the number of assigned labels. The normalized mutual information is 0.3207.

4.2 Experimental procedure

We provide some further explanations regarding our experiments, in addition to what is described in Sect. 3.1.

Train-Test Split of the Datasets along the Time Axis To simulate a real-world citation prediction setting, we split the data on the time axis of the citing documents. This resembles the natural constraint that publications cannot cite other publications that do not exist yet. Given a specific publication year T, we ensure that the training set, $\mathbb{D}_{\text{train}}$, consists of all documents that were published earlier than year T, and use the remaining documents as test data, $\mathbb{D}_{\text{test}}$. Figure 5 shows the distribution of documents over the years along with the split into the training set and test set for PubMed, DBLP, and ACM. Regarding our evaluation, we select the year 2011 for PubMed, 2017 and 2018 for DBLP, and 2014 to 2016 for ACM to obtain a 90:10 ratio between training and test documents.

We also conduct the split between the training and test set along the time axis for the EconBiz and IREON datasets in the subject labeling scenario (Fig. 5). This is challenging because label annotations suffer from concept drift over time (Toepfer & Seifert, 2017). We use the years 2012 and 2013 as test documents for EconBiz and the year 2016 for IREON to obtain a 90:10 train-test ratio, similar to the citation task. Since Reuters contained just two years (1996 and 1997), a time split would have generated too few documents from 1996 to obtain such a ratio. Here, we randomly select 10% of documents for the test set, regardless of the year.

Preprocessing and Dataset Pruning as a Controlled Variable For preprocessing the datasets, we conduct the following three steps: First, we build a vocabulary on the training

Table 3 Characteristics of the citation datasets with respect to different selected pruning thresholds on the minimum item occurrence

Dataset	Pruning	Cited docs	Citations	Documents	Density	α	MI
PubMed	20	20,270	878,359	121,374	0.000357	1.5870	0.4265
	30	8,906	568,563	96,980	0.000658	1.6465	0.3958
	40	4,939	413,746	79,830	0.001049	1.7090	0.3737
	50	3,185	324,693	67,703	0.001506	1.7755	0.3587
DBLP	20	251,405	16,340,121	1,955,132	0.000033	1.5960	0.4879
	30	157,203	13,977,243	1,839,181	0.000048	1.6046	0.4750
	40	109,469	12,271,346	1,742,180	0.000064	1.6127	0.4658
	50	81,817	10,991,096	1,660,462	0.000081	1.6206	0.4591
ACM	20	78,805	5,590,751	786,216	0.000090	1.5840	0.4650
	30	46,782	4,752,086	751,981	0.000135	1.6099	0.4521
	40	31,780	4,189,331	725,635	0.000182	1.6354	0.4435
	50	23,177	3,767,585	702,158	0.000232	1.6610	0.4374

Table 4 Characteristics of the subject label datasets with respect to different selected pruning thresholds on minimum item occurrence

Dataset	Pruning	Classes	Labels	Documents	Density	α	MI
EconBiz	1	4,568	323,670	61,104	0.001160	1.9612	0.2970
	5	3,259	320,048	60,983	0.001610	2.0018	0.2917
	10	2,597	314,738	60,778	0.001994	2.0483	0.2857
	20	1,924	303,693	60,272	0.002619	2.1379	0.2768
IREON	1	10,324	945,888	75,558	0.001213	1.9382	0.1977
	5	6,971	938,677	75,555	0.001782	1.9644	0.1928
	10	5,612	928,701	75,551	0.002190	1.9930	0.1881
	20	4,304	909,156	75,535	0.002797	2.0463	0.1809
Reuters	1	104	2,340,132	744,693	0.030215	148.15	0.3207
	5	104	2,340,132	744,693	0.030215	148.15	0.3207
	10	103	2,340,127	744,693	0.030509	146.71	0.3207
	20	103	2,340,127	744,693	0.030509	146.71	0.3207

set with items cited/assigned more than k times. Second, we filter both the training and test set and retain only items from the vocabulary. Finally, we remove documents with fewer than two of the vocabulary items in their item set.

The pruning threshold k is crucial since it affects both the number of considered items as well as the number of documents. Thus, we control k in our experiments and evaluate the models' performance with respect to its different values. Table 3 shows the characteristics of PubMed, DBLP, and ACM with respect to k, while Table 4 illustrates the same for EconBiz, IREON, and Reuters.

4.3 Baselines

The MLP is not only exploited as a building block of autoencoders but also as an additional baseline model. Furthermore, we use two competitive baselines based on item co-occurrence and singular value decomposition.

Multi-layer Perceptron (MLP) As neural baseline, we use the MLP introduced in Sect. 3.2, which only operates on the documents' metadata. In this case, we optimize the binary cross-entropy BCE($\mathbf{x}, MLP - 2(\mathbf{s})$), where the title and other metadata $\mathbf{s}$ are used as input and citations or subject labels $\mathbf{x}$ as target outputs. With the purpose of a fair comparison, we operate on the same TF-IDF weighted embedded bag-of-words representation (Galke et al., 2017), $\mathbf{s}$, as we have also used to condition the decoder of the autoencoder variants (see Sect. 3.2).

Singular Value Decomposition Singular value decomposition (SVD) factorizes the co-occurrence matrix of items $\boldsymbol{X}^T \cdot \boldsymbol{X}$. Caragea et al. (2013) successfully used SVD for citation recommendation. We include an extended version of SVD in our comparison, which can incorporate title information (Galke et al., 2018). We concatenate the textual features as TF-IDF weighted bag-of-words with the items and perform singular value decomposition on the resulting matrix. To obtain predictions, we only use those indices of the reconstructed matrix that are associated with items.

Item Co-Occurrence As a non-parametric yet strong baseline, we consider the co-citation score (Small, 1973), which is purely based on item co-occurrence. The rationale is that two papers, which have been cited more often together in the past, are more likely to be cited together in the future than papers that were less often cited together. Given training data, $\boldsymbol{X}_{\text{train}}$, we compute the full item co-occurrence matrix $\boldsymbol{C} = \boldsymbol{X}_{\text{train}}{}^T \cdot \boldsymbol{X}_{\text{train}} \in \mathbb{R}^{n \times n}$. At prediction time, we obtain the scores by aggregating the co-occurrence values via matrix multiplication $\boldsymbol{X}_{\text{test}} \cdot \boldsymbol{C}$. On the diagonal of $\boldsymbol{C}$, the occurrence count of each item is retained to model the prior probability.

4.4 Hyperparameter optimization

The hyperparameters are selected by conducting preliminary experiments on PubMed by considering only items that appear 50 or more times in the whole corpus. We chose this scenario because this aggressive pruning results in numbers of distinct items and documents that are similar to the ones of the subject label recommendation datasets. Considering the MLP modules, we conducted a grid search with hidden layer sizes between 50 and 1, 000, initial learning rates between 0.01 and 0.00005, activation functions Tanh, ReLU (Nair & Hinton, 2010), and SELU (Klambauer et al., 2017), along with dropout (Srivastava et al., 2014) (or alpha-dropout in case of SELUs) probabilities between 0.1 and 0.5, and as optimization algorithms a standard stochastic gradient descent and Adam (Kingma & Ba, 2015). For the autoencoder-based models, we considered code sizes between 10 and 500, but only if the size was smaller than the hidden layer sizes of the MLP modules. In the case of adversarial autoencoders, we experimented with Gaussian, Bernoulli, and Multinomial prior distributions, and with linear, sigmoid, and softmax activation on the code layer, respectively.

Although a certain set of hyperparameters may perform better in a specific scenario, we select the following, most robust, hyperparameters: hidden layer sizes of 100 with ReLU (Nair & Hinton, 2010) nonlinearities and drop probabilities of 0.2 after each hidden layer.

The optimization is carried out by Adam (Kingma & Ba, 2015) with initial learning rate 0.001. The autoencoder variants use a code size of 50. We further select a Gaussian prior distribution for the adversarial autoencoder. For SVD, we consecutively increased the number of singular values up to 1, 000. Higher amounts of singular values decreased the performance. We keep this set of hyperparameters across all models and subsequent experiments to ensure a reliable comparison of the models.

4.5 Performance measures

What matters most in the considered scenarios of citation recommendation and subject indexing is identifying a set of relevant items, i.e., completing the set of missing citations on the basis of what is already cited and completing the set of subject labels relevant to a scientific paper, respectively. Given these recommendation scenarios for item set completion, we consider the prediction accuracy as the crucial property to assess.[8] A paper's author or a paper's subject indexer would be most interested in having recommendations for all related papers to be cited and labels to be assigned, respectively. Thus, we focus on prediction accuracy.

We choose the mean reciprocal rank (MRR) as our main evaluation metric of the prediction accuracy. For the evaluation, certain items were omitted on purpose in the test set, as described in Sect. 3.1. For each document, the models ought to predict the omitted items as best as possible. The MRR focuses on the first relevant element of the recommended list (Craswell, 2009) and therefore is a natural choice for the experiments with varying pruning, where only one item is omitted (so only one has to be predicted). It also fits well in the experiments on the influence of the completeness of the citation set and subject labels set, where we incrementally add items. Here, we consider an incremental recommendation scenario where users receive one item at a time and the system should be queried again for the next item. Since in this case we actually need to predict a set of missing items, we additionally compute the mean average precision (MAP). The MAP does not focus only on the first item as the MRR does, yet weights heavily the errors at the top of the list of predicted items and then gradually decreases the significance of the errors for lower items in the list. This ensures rewarding lists of predicted items where the most relevant item are on top. The results with MAP are reported in "Appendix A".

We are given a set of predictions, $\boldsymbol{X}_{\text{pred}}$, for the test set, $\tilde{\boldsymbol{X}}_{\text{test}}$. For each row, we compute the reciprocal rank of the missing items $\boldsymbol{x}_{\text{test}} - \tilde{\boldsymbol{x}}_{\text{test}}$. The reciprocal rank corresponds to one divided by the position of the first omitted item in the sorted set of predictions, $\boldsymbol{x}_{\text{pred}}$. We then average over all documents of the test set to obtain the mean reciprocal rank. To alleviate random effects of model initialization, training data shuffling, and selecting the elements to omit, we conduct three runs for each of the experiments. For a fair comparison, the removed items in the test set remain the same for all models during one run with a fixed pruning parameter.

We also investigate the effect of completeness of the partial input set through the number of dropped elements. We run multiple experiments by dropping different percentages of elements with respect to the size of the original set. We performed experiments for some

[8] Following recommender system terminology, we refer to prediction accuracy as the general property to assess how good predictions are for a user (in contrast to accuracy as a measure), and we specifically focus on measuring the accuracy of rankings of items (Gunawardana & Shani, 2015).

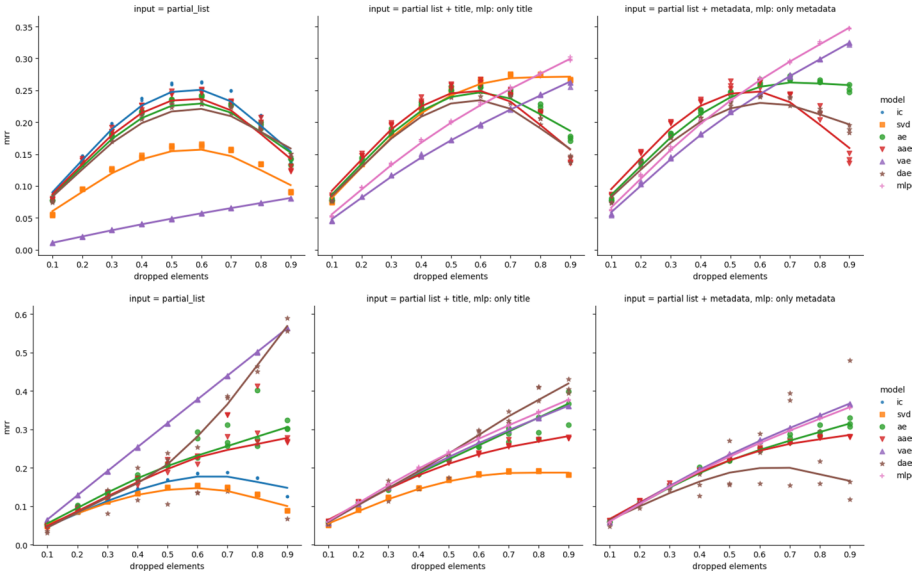


Fig. 6 MRR of predicted citations on the test set with varying number dropped elements for the PubMed (top row) and ACM (bottom row) citation datasets. The minimum item occurrence threshold is set to 55. *Left*: Only the partial set of items is given. *Center*: The partial set of items along with the document title is given. *Right*: The partial set of items is given along with the document title, the authors, the journal title, and the MeSH labels. MLP can only make use of either titles or titles, authors, journal titles, and MeSH labels

given pruning thresholds on PubMed, ACM, and all subject indexing datasets, but we expect a similar behavior for other thresholds.

5 Results

The presentation of the results is structured along the research questions outlined in the introduction. The first question (i), on comparing the recommender performance in two scenarios with different underlying semantics of item co-occurrence, is jointly reflected by Sect. 5.1, which covers the results from citation recommendation task, and Sect. 5.2, which presents the results from the subject label recommendation task. Within each task, we then present the results along with questions (ii) on how the completeness of the partial set of items influences the provided recommendations, (iii) on how pruning affects the models' performance, and (iv) on the influence of using bibliographic metadata as input.

5.1 Results for the citation recommendation datasets

Figure 6 depicts the results for the models on the citation recommendation task with respect to the drop parameter that controls the percentage of dropped elements in the original sets of references, i.e., how the completeness of the partial set of items influences the provided recommendation, on PubMed and ACM, respectively. On PubMed, most of the models peak around a drop threshold of about 50%, independently from whether metadata

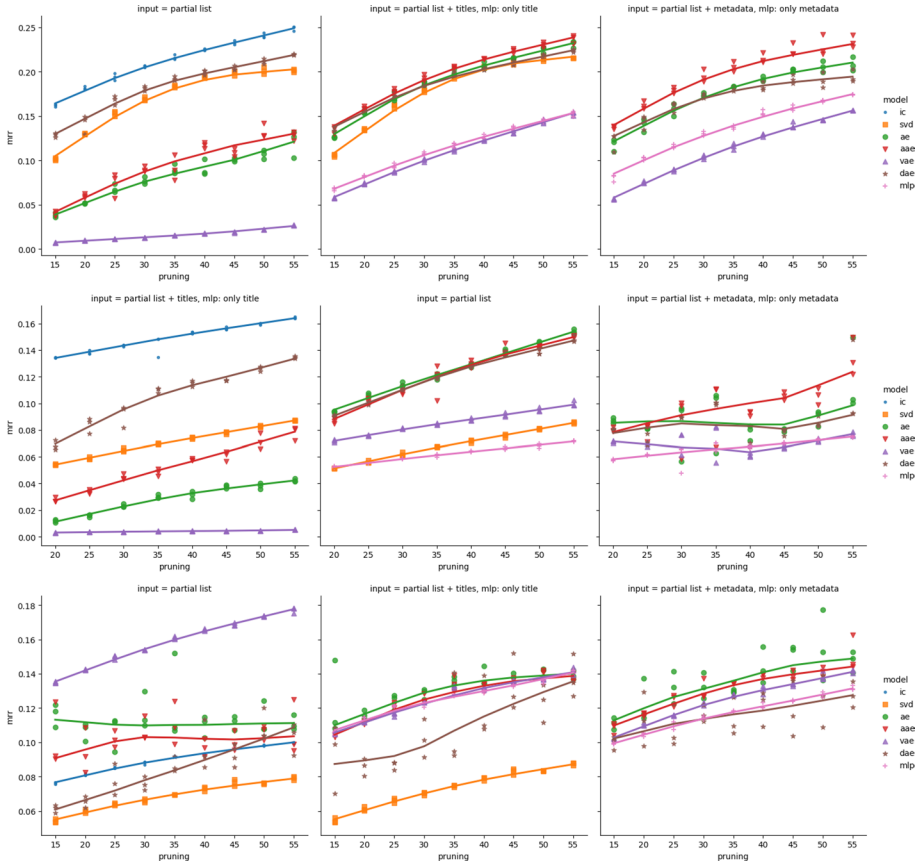


Fig. 7 MRR of predicted citations on the test set with varying minimum item occurrence (pruning) thresholds for the PubMed (top row), DBLP (middle row), and ACM (bottom row) citation datasets. *Left*: Only the partial set of items is given. *Center*: The partial set of items along with the document title is given. *Right*: The partial set of items is given along with the document title, the authors, the journal title, and the MeSH labels (if available for a document). MLP can only make use of either titles or titles, authors, journal titles, and MeSH labels

are used or not. The exceptions are MLP and VAE, which increase also with higher percentages of dropped elements. SVD with titles and AE with more metadata plateau when 60% of elements are dropped. MLP and SVD have low results with few dropped elements but achieve good performance with high drop thresholds. The same holds for VAE when titles or even more metadata are available. On ACM, only item co-occurrence and SVD decrease with more than 60% of elements dropped. However, different autoencoders have a lower improvement with many dropped elements, depending on their type and the additional information used (only the partial set, partial set and titles, or partial set, titles, and more metadata). In contrast, DAE with only the partial set tends to increase more with a drop threshold higher than 50%. The more metadata are used, the lower the improvement with many elements dropped.

Figure 7 shows the results for the models with respect to the pruning parameter that controls the number of considered items as well as the sparsity (see Table 3) on

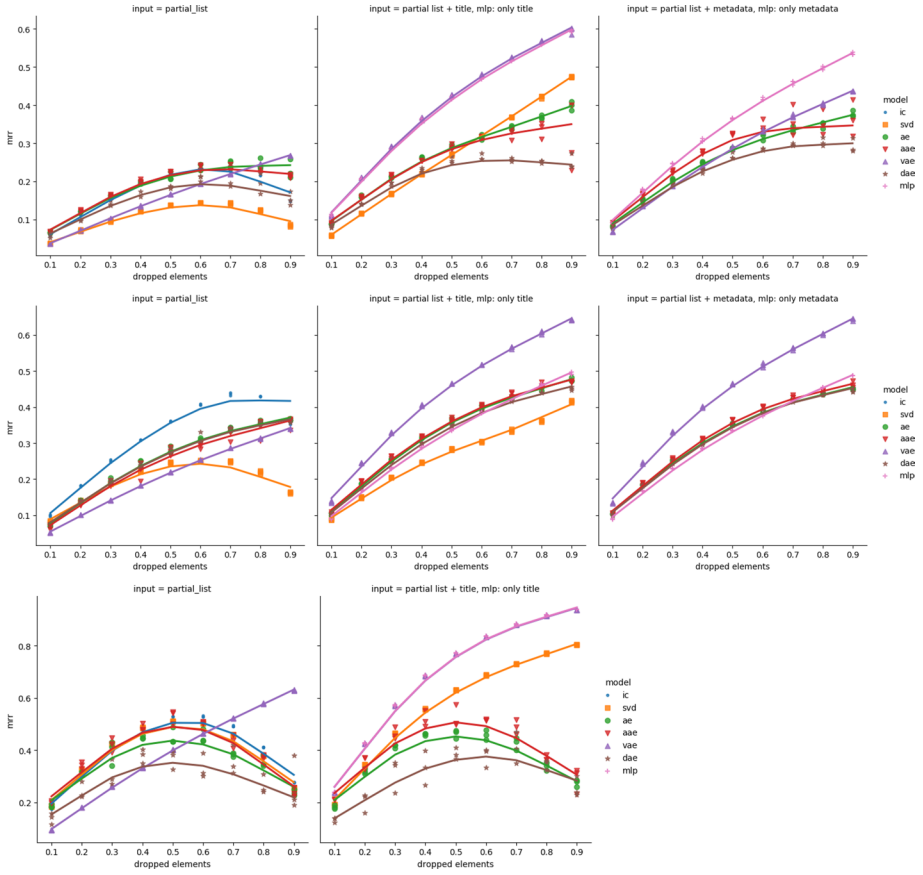


Fig. 8 MRR of predicted subject labels on the test set with varying number of dropped elements for the EconBiz (top row), IREON (middle row), and Reuters (bottom row) datasets. The minimum item occurrence threshold is set to 20. *Left*: Only the partial set of items is given. *Center*: The partial set of items along with the title is given. *Right*: The partial set of items along with the document title and authors is given. MLP can only use either titles or titles and authors

the PubMed, DBLP, and ACM datasets, respectively. We observe a trend that a more aggressive pruning threshold leads to higher scores among all models on all three datasets. However, this phenomenon seems to be more attenuated on the ACM dataset. Notably, on this dataset, AE and AAE seem to be unaffected by the threshold, or even there seems to be a slight decrease for higher thresholds. When no title information is given, the item co-occurrence approach performs best on PubMed and DBLP, while VAE obtained the best scores on ACM. When titles are used, autoencoders (AAE, AE, and DAE) become competitive to the item co-occurrence approach and outperformed other models on PubMed and DBLP. The same holds when additional metadata are available. The results on the ACM dataset show a similar pattern, except that DAE performs worse than VAE. Surprisingly, more metadata yield worse results than titles only.

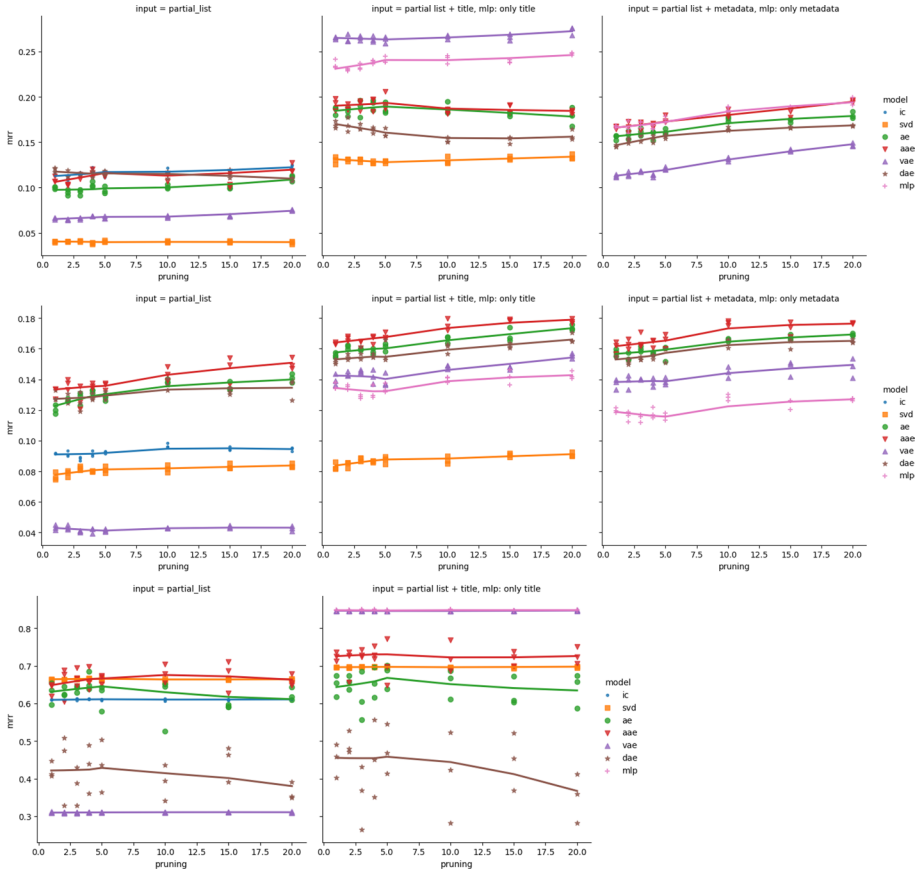


Fig. 9 MRR of predicted subject labels on the test set with varying minimum item occurrence thresholds for the EconBiz (top row), IREON (middle row), and Reuters (bottom row) dataset. *Left*: Only the partial set of items is given. *Center*: The partial set of items along with the document title is given. *Right*: The partial set of items along with the title and authors is given. MLP can only use either titles or titles and authors

5.2 Results for the subject label recommendation datasets

Figure 8 depicts the results for the models for the subject label task with respect to the number of dropped elements on EconBiz, IREON, and Reuters, respectively. As for citations, we performed experiments only for some given pruning thresholds, but we expect a similar behavior for other thresholds. When no title information is available, most of the models peak or plateau around a drop threshold of 50% and 60% on Reuters and EconBiz, respectively. On IREON, item co-occurrence plateaus at 70%, SVD peaks at 60%, the other models show a steady increase. With titles only, SVD performs poorly when few elements are dropped, while outperforms various models with many elements dropped (except for IREON). When titles and additional metadata are given, only AE, AAE, and DAE increase considerably less with many elements dropped on EconBiz, while they peak at about 50% on Reuters. On IREON, all the models suffer less when many elements are dropped. VAE and MLP are generally the best performing models in all datasets, although the most effective varies with the dataset and the metadata used.

Figure 9 shows the results for the models with respect to the pruning parameter that controls the number of considered items and the corresponding sparsity (see Table 4) on the EconBiz, IREON, and Reuters datasets, respectively. When no title information is available, autoencoders (but VAE) are competitive to the item co-occurrence approach. When titles are given, the models achieve considerably higher scores than all models operating without this information. Specifically, VAE achieves the best results, closely followed by MLP, on EconBiz, while AAE is the best-performing model on IREON. On Reuters, MLP and VAE achieve the same performance. Similar to the citation task, using additional metadata decreases the performance compared to using only titles, although in IREON the decline is notably lower. VAE is particularly negatively affected by more metadata on EconBiz. Reuters only provides titles, so it was impossible to use additional metadata.

6 Discussion

We first recall the main findings and compare the different item co-occurrence semantics in the two tasks (research question i). Subsequently, for each task, we discuss each of the remaining research questions (ii to iv).

6.1 Main findings

The two tasks of citation recommendation and subject label recommendation may seem rather similar at a first glance, but they are actually not. Regarding research question (i), our experiments show that what is already cited is much more informative than which subject labels are already assigned. This is supported by the mutual information, which is greater than 0.5 for the citation datasets, but below or close to 0.3 for the subject label datasets. In the citation recommendation task, where item co-occurrence implies relatedness, the approach based on the co-occurrence is usually a strong baseline. VAE (using only item sets) outperforms this baseline on ACM, but performs worse when titles and other metadata are exploited. In the subject label recommendation task, where co-occurrence of items implies diversity, the best method depends on the dataset: AAE in IREON, VAE in EconBiz (followed by MLP), and MLP and VAE (with titles) in Reuters, although item co-occurrence may still achieve good performance in IREON. Instead, DAE are less stable as its results highly vary among multiple runs on the same dataset.

Concerning research question (ii), our experiment on the varying number of dropped elements indicates that the models which do not show the boomerang-shaped curve when using metadata are the ones that rely more on titles and other metadata than on the partial set of items. This enables them to perform well also when very few items are given as input in both the considered tasks.

Regarding research question (iii), it is desirable to have low pruning thresholds to avoid a “rich get richer” phenomenon, where documents highly cited, or labels frequently assigned, are privileged. In fact, highly cited documents, or often used subject labels, are also the ones most likely to be known, while recommendations should also lead to discovering previously unknown items.

Finally, about research question (iv), our results show that, counter-intuitively, it is not always better to add more metadata as input. Usually, titles are helpful but sometimes models which rely solely on the partial set of items are the top performers. Titles are the most

useful metadata field in our experiments. Adding further metadata not only typically did not improve the performance, but sometimes even decreased them.

6.2 Comparison of different semantics of item co-occurrence

We observe different relationships between the factors of (i) type of recommendation task and the performance with (ii) varying completeness of the partial set, (iii) degree of sparsity, as well as (iv) input modalities. We discuss the results per type of recommendation task in Sects. 6.3 and 6.4.

Regarding the research question (ii), the recommendation performance depends a lot on the completeness of the partial set. Most models perform best having around 50% of items from the original set, but VAE and MLP achieve good performance also with 90% of elements dropped. It is noteworthy that pruning almost never changes the order of the top performers, while the top performers differ by varying the number of dropped elements.

Second, on research question (iii), by applying several thresholds on minimum item occurrence, we controlled the number of considered items and thus the degree of sparsity. The result is that all considered models are similarly affected by the increased sparsity and difficulty caused by higher numbers of considered items. Unsurprisingly, the results decrease with lower pruning thresholds, as pruning can heavily influence the results (Beel et al., 2016). Interestingly, there are differences between the two tasks: while the decrease is pronounced in the citation datasets, it is very low in the other ones. We think that this decrease is low in the subject label datasets because they are considerably less sparse (see Table 4 vs Table 3). Differences persist even when datasets are similar. EconBiz has 4, 568 classes without pruning and PubMed has 4, 939 cited documents with pruning at 40 and 3, 904 with pruning at 45. On PubMed (citations), the best MRR is about 21% with item co-occurrence (not improving with metadata), while on EconBiz (subject labels), all models yield similar results only when using metadata, and without metadata are below 13%.

Finally, regarding research question (iv), on the citation task, the partial set of citations is the most important information to recommend potentially missing citations. For the subject label recommendation task, however, the MLP model, which exploits titles only, achieves the best performance in one dataset and is generally competitive. On the citation recommendation task, autoencoders and SVD become competitive to the strong co-citation baseline when titles are used.

6.3 Detailed discussion of the citation recommendation task

The number of elements dropped in the partial input sets differently affects the models. Most of the models improve until about 50% of elements are dropped, then their performance decreases. In fact, the more elements are dropped the less information is used as input, but the task becomes also easier as there is more than one correct answer (one document or label to predict). When metadata are provided together with the partial set, the performance decrease is lower. As shown in "Appendix A", this is particularly the case when the partial input set is small (many elements are removed) since metadata compensates for fewer items in the partial set. For the citation recommendation task, where item co-occurrence implies relatedness, the partial set of citations is usually most important. However, for the PubMed dataset, the best results are achieved using metadata. The improvement is largely due to the MeSH labels, as an additional analysis shows (see "Appendix A"). Since PubMed was the only citation dataset providing subject labels, we could not verify if labels

would help also in other datasets and further investigation on their effect is necessary. We expect a similar effect on other datasets because the general subjects of a paper are typically related to what is cited and what is not.

The MLP results improve also with many elements dropped as it does not exploit the partial set of items. VAE shows similar behavior. The latter is capable of providing good predictions even when few items are given as input. The reason could be the generalization provided through the Gaussian prior to the code. The latent representations learned by distinguishing the code from a smooth prior make the model more robust to sparse input vectors because smoothness is key for good representations that disentangle the explanatory factors of variation (Bengio et al., 2013). Although both VAE and AAE impose a Gaussian prior on the code, VAE is better in some cases. A possible explanation is that VAE is more stable during optimization (Tolstikhin et al., 2018). We studied the effect of a larger partial set of items as input on the results when many elements were dropped. This enables us to discriminate whether the reason for VAE and MLP not decreasing in performance was the larger amount of training data or the fact that these methods were better at separating relevant documents from noise. In order to do so, we ran additional experiments in PubMed, where 80% of elements were dropped from the item set. We then compared results using only documents with either long or short sets of citations as input (see "Appendix A"). All the models benefit from a larger partial set without metadata. This suggests that VAE and MLP can better separate relevant documents from noise. However, the best performance is achieved with metadata and shorter partial sets as input.

On ACM, the results continue to improve even with very few items in the partial set (i. e., many elements are dropped), except for item co-occurrence and SVD. This may be due to ACM characteristics. The mutual information for ACM (0.5282) is the lowest among the citation datasets (0.5407 for DBLP and 0.5996 for PubMed). This means that already cited documents are less informative in ACM compared to the other two datasets.

Regarding the results for the experiments with varying pruning, co-citation is known to be an important baseline (Small, 1973). However, we have shown that is not always the case. For the ACM dataset, VAE is the best method. This is interesting, because in other cases VAE particularly underperforms on DBLP and PubMed, especially without metadata. This may be due to ACM's special characteristics, as already discussed for the number of dropped elements. Specifically, already cited documents are less informative in ACM compared to the other two datasets due to the lower mutual information.

On the influence of metadata, MLP and autoencoders have the advantage of exploiting metadata information (autoencoders along with the partial set of citations), which improves the results. The use of titles is most effective. The use of other metadata attributes, such as authors and venues, on top of titles, does not improve the recommendation performance. From the partial input set, autoencoders can learn co-occurrence within item sets and may also learn to put appropriate weights in the bias parameters, if it is helpful for the overall objective. In this task of citation recommendation, the co-occurrence of documents within item sets is of great importance as related papers are typically cited together, which explains the strength of the item co-occurrence baseline. Although MLP can also learn bias in the output layer to model the items' prior distribution, autoencoders can model the relation between cited documents from the partial input set. MLP cannot model the relation between cited documents because it only uses the titles as input and not the partial set. When considering different types of metadata in the case of MLP, using titles is generally as effective as, or even more effective than, using additional metadata available (together with the titles). We think this is because the title is more indicative than other metadata when deciding whether to cite a paper. Although researchers may tend to cite more often

some papers from well-known venues as well as some authors, because of similarity in topics or because they know them, the title has a stronger relation to the paper than the authors and venues. Furthermore, the advantage of the title is that this metadata is always available, even in the news domain, while other metadata are not. For example, in PubMed only about 77% of the documents have MeSH labels, in ACM roughly 93% of documents include authors, and in DBLP around 83% of papers hold the venue (see Table 2), while for all documents there is a title.

6.4 Detailed discussion of subject label recommendation task

For the subject label recommendation task, the number of elements dropped has a lower effect on the models compared to the citation recommendation task. Most of the models improve until when about 50% of the elements in the original set are dropped, then their performance plateaus or slightly decreases. Nevertheless, some models can provide good predictions also with few elements in the partial set. As in the citation recommendation task, results of MLP and VAE improve even with many elements dropped. Only with the Reuters dataset, many models show a notable decline. This could be due to its distribution of the label occurrence, which is the only one that does not follow the power-law distribution. On the contrary, there is a low number (roughly 100 compared to 4, 600 in EconBiz and 10, 000 in IREON) of fairly well-balanced labels to choose from (see Fig. 4). Another reason could be that in the Reuters dataset the labels have no hierarchy, contrary to the other two subject label datasets that are based on a professional taxonomy. Subject indexers usually assign the ancestor instead of the child subjects when two or more subject labels with a common ancestor in the hierarchical thesaurus match (Große-Bölting & Scherp, 2015). Thus, two subjects that are semantically related because they share a common ancestor are unlikely to co-occur in the annotations of a single document. Without a hierarchy, different subject labels can be similar and no common ancestor is available to use instead. For example, the news "*Clinton signs law raising minimum wage to $5.15*" is assigned the subjects *EMPLOYMENT/LABOUR*, *LABOUR*, and *LABOUR ISSUES*, which are rather similar. Finally, as the main purpose of subject indexing is retrieval, the fact that there are many labels used, often suggests that recall is preferred over precision. While in a library it may be desirable to retrieve fewer results but all highly relevant to a query, in the news domain it may be best to retrieve as many results as possible, although some could be less relevant.

In the experiments with varying pruning, MLP and VAE achieve the highest mean reciprocal rank in two out of three datasets. Thus, already assigned subjects are less informative for a subject label recommendation task than the titles are. Two subjects that are semantically related because they share a common ancestor is unlikely to co-occur in the annotations of a single document since subject indexers tend to rely on the ancestor instead of the child subjects. On the IREON dataset, instead, autoencoders and notably AAE, are most effective. A reason might be that IREON has much more different subject labels than the other datasets (see Table 4). Therefore, co-occurrence statistics, which are modeled by AE variants, might be more informative.

Regarding the influence of metadata, adding other metadata with titles generally decreases the performance. As for citations, other metadata may be less indicative of the paper content. Furthermore, only the authors are available in addition to titles. Authors' names are present in roughly 98% and 83% of documents in the EconBiz and IREON datasets, respectively. VAE is the best model (together with MLP) in Reuters, but performs

poorly without titles. This suggests that, when provided with titles, VAE learns to ignore the input item set, which is not possible with only the partial set of items as input.

6.5 Threats to validity

The availability and occurrences of metadata fields vary among the datasets, as shown in Table 2. Citations datasets have more metadata attributes available than the subject label datasets. Titles and authors can be used in all three citation datasets; in PubMed also journal and MeSH labels can be exploited, and venues are present in ACM and DBLP. For the subject label datasets, only titles and authors can be used, apart from Reuters, which offers only titles. So it was only partially possible to use the same or similar set of metadata fields among the datasets. Notably, we could not use abstracts because they were always or often missing. Only ACM and DBLP contain such information but it is rarely available in ACM (less than 4%), so we decided not to use it in our experiments.

We tested the models' performance and the impact of different semantics of item co-occurrence, the completeness of partial input set, the pruning, and metadata on various datasets, for each of the two recommendation tasks. As the results are consistent among the datasets, we have good reasons to assume that the results can also be generalized to other, similar datasets in the considered domains. Our method to add metadata is general and can handle different types of metadata. Thus, the models can also be applied to similar tasks in other domains, which could also benefit from the use of metadata. For example, we have previously shown that metadata are beneficial for automatic playlist continuation (Vagliano et al., 2018).

6.6 Practical implications

Our experiments have a high practical relevance as they are close to real-world settings. This comes in three aspects. First, we use six real-world datasets from five different domains. Thus, the experimental results show how the recommender models would behave in real-world contexts. Second, the splitting of the documents in training and test set along the time axis resembles the natural constraint that newly written publications can only cite already published works and only papers published before are already annotated. Additionally, applying this chronological split to subject labels also accounts for concept drift (Webb et al., 2018). Third, by taking into account the typical long-tail distribution in users' feedback to items (in our case, citations and label annotations), we have also investigated performance with low pruning thresholds, i. e., including items with few citations and labels rarely used to annotate documents. This further strengthens the reliability of our experimental results in real-world settings, in contrast to existing studies with limited datasets induced by pruning items based on frequency.

7 Conclusion

Different semantic interpretation of item co-occurrence in recommendation tasks highly affects the preferable input modalities. When item co-occurrence resembles relatedness, such as in citations, the set of already cited documents is usually most beneficial. However, metadata may help with short partial item sets: for instance, using subject labels as input for the citation task was particularly effective in PubMed, the

only dataset that provided them, when many elements were dropped from the item set. In subject recommendations, co-occurring subject labels do not imply that these subjects are similar. Instead, a document's actual research subject needs to be described by using multiple, diverse subject labels as annotations. Incorporating multiple input modalities offers a conceptual benefit. However, adding more metadata is not always useful, and relying on just titles or the partial set of items can be more effective. Further analysis on using subject labels as input metadata for citation recommendation should be conducted. All of the evaluated methods are similarly sensitive to data sparsity, but variational autoencoders and multi-layer perceptron are more robust with few items in the partial set. This is likely because they rely more on titles and other metadata than on the initial item set.

When addressing a recommendation task, the semantics of item co-occurrence and the completeness of the partial set of items should be considered to decide whether the partial list of items should be supplied to a recommendation model as input. Regarding the two scenarios considered, we can state: In citation recommendation, where co-occurring items are similar, the models can perform well without using additional metadata. In subject indexing, where co-occurring items are diverse, using the content is more effective than using the partial item set.

As future work, we plan to further study whether subject labels used as input for citation recommendation can outperform the partial list of items and to use additional metadata and content, such as the abstract. We also intend to investigate further recommendation scenarios with respect to the influence of the semantics of item co-occurrence on the recommendation performance. Our results show that different models are most effective with low, medium, or many items in the partial set. Thus, an interesting future direction could be using ensembles of models. Combining different models could be useful for handling different numbers of either cited documents or already assigned subject labels as input.

Appendix A: A detailed analysis of the PubMed dataset

We investigated the effect of using MeSH labels as metadata on PubMed. This is the only citation dataset that provides subject labels, which can be used as additional input information for our models. Figure 10 shows the results. While using MeSH labels improves the performance, the models are similarly affected by the drop threshold in both cases.

We also studied the effect of a larger partial set of items as input on the results. In our experiments, MLP and VAE do not drop in their performance when few candidates are available (higher values of drop ratio). In order to discriminate if the reason for these results is because of larger training or because these methods are more useful in separating relevant documents from noise, we split the documents based on the median number of references as input. This allows us to compare results with longer or shorter lists of references as training input. As Table 5 shows, all the models benefit from a longer partial list without metadata, but better performance is achieved using metadata. Metadata are particularly helpful when a shorter partial list is provided as input.

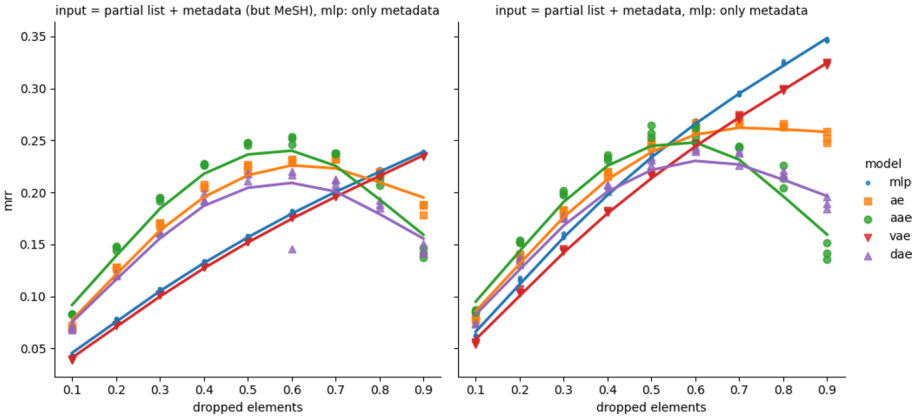


Fig. 10 MRR of predicted citations on the test set with varying number dropped elements for the PubMed citation dataset. The minimum item occurrence threshold is set to 55. *Left*: The partial set of items is given along with the document title, the authors, and the journal title. *Right*: The partial set of items is given along with the document title, the authors, the journal title, and the MeSH labels

Table 5 MRR of predicted citations for PubMed using either only documents with fewer references than the median number of references or with more references than the median number of references

Model	< 30 references		≥30 references	
	Partial list	Metadata	Partial list	Metadata
IC	.1958 (.0019)	–	.2136 (.0004)	–
SVD	.1409 (.0004)	–	.1503 (.0002)	–
AE	.1734 (.0007)	.2946 (.0013)	.1996 (.0076)	.2520 (.0015)
AAE	.1485 (.0023)	.2756 (.0013)	.1957 (.0084)	.2256 (.0196)
VAE	.1337 (.0004)	.3580 (.0010)	.0907 (.0002)	.2750 (.0009)
DAE	.1311 (.0002)	.2710 (.0024)	.2022 (.0100)	.2049 (.0011)
MLP	**–**	**.3633 (.0014)**	–	**.2926 (.0011)**

The best results are reported in bold

Appendix B: Mean average precision results

In addition to the mean reciprocal rank (MRR), we also assess the mean average precision (MAP) for the experiments investigating the effect of the number of dropped elements. Results are provided in Figs 11 (citations), 12 (subject labels), and Table 6. The models show substantially the same behavior with MAP as with MRR. The only exception is that the VAE demonstrates an overall lower increase of performance with an increasingly higher number of dropped elements for the subject indexing datasets.

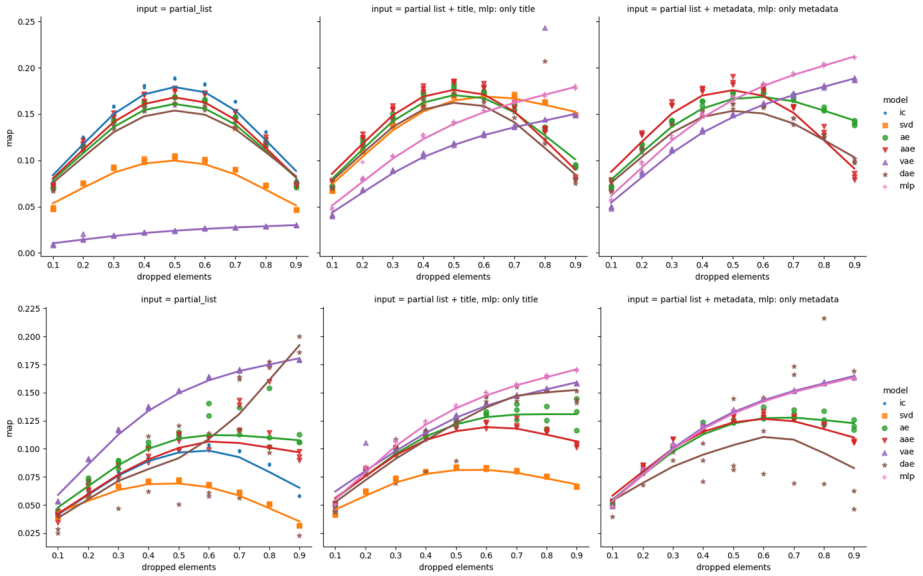


Fig. 11 MAP of predicted citations on the test set with varying number dropped elements for the PubMed (top row) and ACM (bottom row) citation datasets. The minimum item occurrence threshold is set to 55. *Left*: Only the partial set of items is given. *Center*: The partial set of items along with the document title is given. *Right*: The partial set of items is given along with the document title, the authors, the journal title, and the MeSH labels. MLP can only make use of either titles or titles, authors, journal titles, and MeSH labels

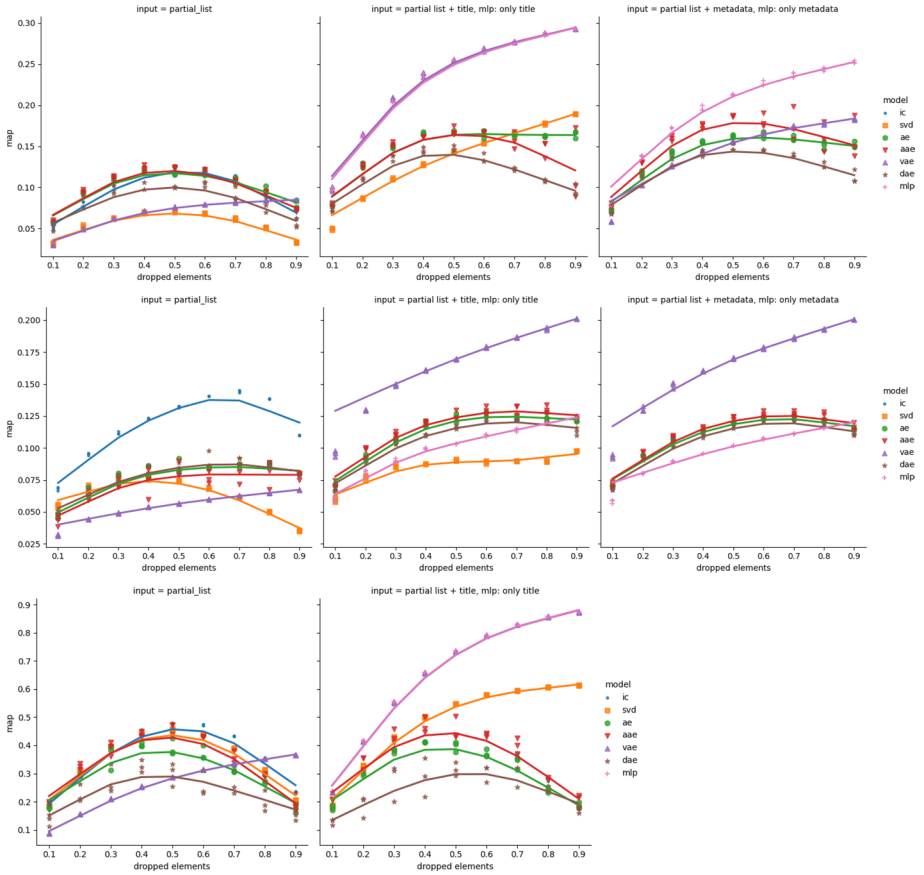


Fig. 12 MAP of predicted subject labels on the test set with varying number of dropped elements for the EconBiz (top row), IREON (middle row), and Reuters (bottom row) datasets. The minimum item occurrence threshold is set to 20. *Left*: Only the partial set of items is given. *Center*: The partial set of items along with the title is given. *Right*: The partial set of items along with the document title and authors is given. MLP can only use either titles or titles and authors

Table 6 MAP of predicted citations for PubMed using either only documents with fewer references than the median number of references (left column) or with more references than the median number of references (right column)

Model	< 30 references		≥30 references	
	Partial list	Metadata	Partial list	Metadata
IC	.1641 (.0015)	–	.1321 (.0006)	–
SVD	.1165 (.0009)	–	.0838 (.0004)	–
AE	.1205 (.0006)	.2299 (.0011)	.1199 (.0022)	.1503 (.0007)
AAE	.0902 (.0011)	.2250 (.0007)	.1187 (.0019)	.1574 (.0237)
VAE	.0795 (.0001)	.2767 (.0009)	.0384 (.0002)	.1670 (.0003)
DAE	.0808 (.0001)	.2076 (.0018)	.1190 (.0053)	.1200 (.0011)
MLP	**–**	**.2818 (.0017)**	–	**.1818 (.0009)**

The best results are reported in bold

Acknowledgements We thank Gunnar Gerstenkorn for his support with the data preparation and preprocessing.

Funding This work was partially supported by the EU H2020 project MOVING (contract no 693092).

Availability of data and material No new data were generated in support of this research. The analyzed data underlying this article from the used PubMed, ACM, DBLP, EconBiz, and Reuters datasets are available in their own repository or website. The PubMed dataset is available on the CITREC website at isg.uni-konstanz.de/projects/citrec; ACM and DBLP are available on the AMiner website at aminer.org/citation; EconBiz data are available as a Kaggle dataset at kaggle.com/hsrobo/titlebased-semantic-subject-indexing; Reuters is stored in the datahub repository at old.datahub.io/dataset/rcv1-v2-lyrl2004. IREON data were provided by the German Information Network for International Relations and Area Studies by permission. No further material was used apart from the code which information is provided next.

Code availability The source code for reproducing and extending our experiments is openly available at github.com/lgalke/aae-recommender.

Declarations

Conflicts of interest We have no conflicts of interest to report.

References

Ali, Z., Ullah, I., Khan, A., Ullah Jan, A., & Muhammad, K. (2021). An overview and evaluation of citation recommendation models. Scientometrics.

Ali, Z., Kefalas, P., Muhammad, K., Ali, B., & Imran, M. (2020). Deep learning in citation recommendation models survey. *Expert Systems with Applications, 162,* 113790.

Bai, J., & Ban, Z. (2019). Collaborative multi-auxiliary information variational autoencoder for recommender systems. In *ICMLC* (pp. 501–505). ACM.

Barbieri, J., Alvim, L. G. M., Braida, F., & Zimbrão, G. (2017). Autoencoders and recommender systems: COFILS approach. *Expert Systems with Applications, 89,* 81–90.

Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). paper recommender systems: A literature survey. *International Journal on Digital Libraries, 17*(4), 305–338.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *TPAMI, 35*(8), 1798–1828.

Bonnin, G. & Jannach, D. (2014). Automated generation of music playlists: Survey and experiments 47 (2).

Boughareb, D., Khobizi, A., Boughareb, R., Farah, N., & Seridi, H. (2020). A graph-based tag recommendation for just abstracted scientific articles tagging. *International Journal of Cooperative Information Systems, 29*(3), 2050004:1-2050004:30.

Cao, S., Yang, N., & Liu, Z. (2017). Online news recommender based on stacked auto-encoder. In *ICIS* (pp. 721–726). IEEE.

Caragea, C., Silvescu, A., Mitra, P., & Giles, C. L. (2013). Can't see the forest for the trees?: a citation recommendation system. In *JCDL* (pp. 111–114). ACM.

Chen, Y., & de Rijke, M. (2018). A collective variational autoencoder for top-n recommendation with side information. In *DLRS@RecSys* (pp. 3–9). ACM.

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: The state of the art. *User Modeling and User-Adapted Interaction, 25*(2), 99–154.

Chen, H., Yang, Y., Lu, W., & Chen, J. (2020). Exploring multiple diversification strategies for academic citation contexts recommendation. *Electron Libre, 38*(4), 821–842.
Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*. Wiley.
Craswell, N. (2009). *Mean reciprocal rank* (p. 1703). Springer.
Cucchiarelli, A., Morbidoni, C., Stilo, G., & Velardi, P. (2019). A topic recommender for journalists. *Information Retrieval Journal, 22*(1–2), 4–31.
Dacrema, M. F., Cremonesi, P., & Jannach, D. (2019). Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *RecSys* (pp. 101–109). ACM.
Ebesu, T., & Fang, Y. (2017). Neural citation network for context-aware citation recommendation. In *SIGIR* (pp. 1093–1096). ACM.
Färber, M., & Jatowt, A. (2020). Citation recommendation: Approaches and datasets. *International Journal on Digital Libraries, 21,* 375–405.
Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., & Reiterer, S. (2013). *Toward the next generation of recommender systems: Applications and research challenges* (pp. 81–98). Springer.
Galke, L., Mai, F., Schelten, A., Brunsch, D., & Scherp, A. (2017). Using titles vs. full-text as source for automated semantic document annotation. In *K-CAP* (pp. 20:1–20:4). ACM.
Galke, L., Mai, F., Vagliano, I., & Scherp A. (2018). Multi-modal adversarial autoencoders for recommendations of citations and subject labels. In *UMAP* (pp. 197–205). ACM.
Galke, L., Saleh, A., & Scherp, A. (2017). Word embeddings for practical information retrieval. In *GI-Jahrestagung, GI* (pp. 2155–2167).
Gipp, B., Meuschke, N., & Lipinski, M. (2015). CITREC: An evaluation framework for citation-based similarity measures based on TREC genomics and PubMed Central. In *iConference*.
Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., & Bengio, Y. (2014). Generative adversarial nets. In *NIPS* (pp. 2672–2680).
Große-Bölting, G., Nishioka, C., & Scherp, A. (2015). A comparison of different strategies for automated semantic document annotation. In *K-CAP* (pp. 8:1–8:8). ACM.
Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *KDD* (pp. 855–864). ACM.
Gunawardana, A., & Shani, G. (2015). *Evaluating recommender systems* (pp. 265–308). Springer.
Hamilton, W. L. (2020). Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 14*(3), 1–159.
He, M., Meng, Q., & Zhang, S. (2019). Collaborative additional variational autoencoder for top-n recommender systems. *IEEE Access, 7,* 5707–5713.
Huang, W., Kataria, S., Caragea, C., Mitra, P., Giles, C. L., & Rokach, L. (2012). Recommending citations: Translating papers into references. In *CIKM* (pp. 1910–1914). ACM.
Huang, W., Wu, Z., Chen, L., Mitra, P., & Giles, C. L. (2015). A neural probabilistic model for context based citation recommendation. In *AAAI* (pp. 2404–2410.
Hu, L., Li, C., Shi, C., Yang, C., & Shao, C. (2020). Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing and Management, 57*(2), 102–142.
ISO 999. (1996). Information and documentation—Guidelines for the content. organization and presentation of indexes.
Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR*. OpenReview.net.
Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In *ICLR*. OpenReview.net.
Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. In *NIPS* (pp. 972–981).
Kumar, S., Zhang, X., & Leskovec, J. (2019). Predicting dynamic embedding trajectory in temporal interaction networks. In *KDD* (pp. 1269–1278). ACM.
Lei, K., Fu, Q., Yang, M., & Liang, Y. (2020). Tag recommendation by text classification with attention-based capsule network. *Neurocomputing, 391,* 65–73.
Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Machine Learning Research, 5*.
Li, X., & She, J. (2017). Collaborative variational autoencoder for recommender systems. In *KDD* (pp. 305–314). ACM.
Li, S., Kawale, J., & Fu, Y. (2015). Deep collaborative filtering via marginalized denoising auto-encoder. In *CIKM* (pp. 811–820). ACM.
Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara T. (2018). Variational autoencoders for collaborative filtering. In *WWW* (pp. 689–698). ACM.
Liu, Y., Wang, S., Khan, M. S., & He, J. (2018). A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering. *Big Data Mining and Anyalytics, 1*(3), 211–221.

Lops, P., de Gemmis, M., & Semeraro, G. (2011). *Content-based recommender systems: State of the art and trends* (pp. 73–105). Springer.
Mai, F., Galke, L., & Scherp, A. (2018). Using deep learning for title-based semantic subject indexing to reach competitive performance to full-text. In *JCDL* (pp. 169–178). ACM.
Majumdar, A., & Jain, A. (2017). Cold-start, warm-start and everything in between: An autoencoder based approach to recommendation. In *IJCNN* (pp. 3656–3663). IEEE.
Makhzani, A., Shlens, J., Jaitly, N., & Goodfellow, I. J. (2015). Adversarial autoencoders. CoRR arxiv:1511.05644 (there is no conference version).
Ma, S., Zhang, C., & Liu, X. (2020). A review of citation recommendation: From textual content to enriched context. *Scientometrics, 122*(3), 1445–1472.
McNee, S. M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S. K., Rashid, A. M., Konstan, J. A., & Riedl, J. (2002). On the recommending of citations for research papers. In *CSCW* (pp. 116–125). ACM.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *NIPS* (pp. 3111–3119).
Musto, C., Basile, P., Lops, P., de Gemmis, M., & Semeraro, G. (2017). Introducing linked open data in graph-based recommender systems. *Information Processing and Management, 53*(2), 405–435.
Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *ICML* (pp. 807–814). Omnipress.
Nam, J., Kim, Y., Loza Menc'ia, E., Park, S., Sarikaya, R., & Fürnkranz, J. (2019). Learning context-dependent label permutations for multi-label classification. In *ICML, PMLR* (pp. 4733–4742).
Nam, J., Loza Mencía, E., Kim, H. J., & Fürnkranz J. (2017). Maximizing subset accuracy with recurrent neural networks in multi-label classification. In *NIPS* (pp. 5419–5429).
Newman, M. E. (2005). Power laws, pareto distributions and zipf's law. *Contemporary Physics, 46*(5).
Pan, Y., He, F., & Yu, H. (2020). Learning social representations with deep autoencoder for recommender system. *World Wide Web, 23*(4), 2259–2279.
Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *KDD* (pp. 701–710). ACM.
Posch, L., Wagner, C., Singer, P., & Strohmaier, M. (2013). Meaning as collective use: Predicting semantic hashtag categories on twitter. In *WWW* (pp. 621–628). ACM.
Raamkumar, A. S., Foo, S., & Pang, N. (2017). Using author-specified keywords in building an initial reading list of research papers in scientific paper retrieval and recommender systems. *Information Processing and Management, 53*(3), 577–594.
Rezende, D. J., . Mohamed, S, & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *ICML* (pp. 1278—1286). JMLR.org.
Sakib, N., Ahmad, R. B., & Haruna, K. (2020). A collaborative approach toward scientific paper recommendation using citation context. *IEEE Access, 8,* 51246–51255.
Sedhain, S., Menon, A. K., Sanner, S., Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. In *WWW* (pp. 111–112). ACM.
Sigurbjörnsson, B., & van Zwol, R. (2008). Flickr tag recommendation based on collective knowledge. In *WWW* (pp. 327–336). ACM.
Silva, N., Carvalho, D., Pereira, A. C. M., Mourão, F., & da Rocha, L. C. (2019). The pure cold-start problem: A deep study about how to conquer first-time users in recommendations domains. *Information Systems, 80,* 1–12.
Small, H. (1973). Co-citation in the scientific literature: A new measure of the relationship between two documents. *JASIS, 24*(4), 265–269.
Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *JMLR, 15*(1), 1929–1958.
Steck, H. (2019). Embarrassingly shallow autoencoders for sparse data. In *WWW* (pp. 3251–3257). ACM.
Strub, F., Gaudel, R., & Mary, J. (2016). Hybrid recommender system based on autoencoders. In *DLRS@ RecSys* (pp. 11–16). ACM.
Sun, J., Zhu, M., Jiang, Y., Liu, Y., & Wu, L. (2021). Hierarchical attention model for personalized tag recommendation. *Journal of the Association for Information Science and Technology, 72*(2), 173–189.
Tang, L., Rajan, S., & Narayanan V. K. (2009). Large scale multi-label classification via metalabeler. In *WWW* (pp. 211–220). ACM.
Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008). Arnetminer: Extraction and mining of academic social networks. In *KDD* (pp. 990–998). ACM.
Tao, S., Shen, C., Zhu, L., & Dai, T. (2020). SVD-CNN: A convolutional neural network model with orthogonal constraints based on SVD for context-aware citation recommendation. *Computational Intelligence and Neuroscience, 2020,* 1–12.

Toepfer, M., & Seifert, C. (2017). Descriptor-invariant fusion architectures for automatic subject indexing. In *JCDL* (pp. 31–40). IEEE.
Tolstikhin, I. O., Bousquet, O., Gelly, S., & Schölkopf, B. (2018). Wasserstein auto-encoders. In *ICLR*. OpenReview.net.
Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining, 3*(3), 1–13.
Vagliano, I., Galke, L., Mai, F., & Scherp A. (2018). Using adversarial autoencoders for multi-modal automatic playlist continuation. In *Recommender systems challenge* (pp. 5:1–5:6). ACM.
Vagliano, I., Monti, D., Scherp, A., & Morisio M. (2017). Content recommendation through semantic annotation of user reviews and linked data. In *K-CAP* (pp. 32:1–32:4). ACM.
Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *ICML* (pp. 1096–1103). ACM.
Vinyals, O., Bengio, S., & Kudlur, M. (2016). Order matters: Sequence to sequence for sets. In *ICLR*. OpenReview.net.
Wang, H., Wang, N., & Yeung, D. (2015). Collaborative deep learning for recommender systems. In *KDD* (pp. 1235–1244). ACM.
Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y., Wang, B., Zhang, P., & Zhang, D. (2017). Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *SIGIR* (pp. 515–524). ACM.
Wang, D., Deng, S., & Xu, G. (2018). Sequence-based context-aware music recommendation. *Information Retrieval Journal, 21*(2–3), 230–252.
Webb, G. I., Lee, L. K., Goethals, B., & Petitjean, F. (2018). Analyzing concept drift and shift from sample data. *Data Mining and Knowledge Discovery, 32*(5), 1179–1199.
Wouters, P. F. (1999). The citation culture, Ph.D. thesis, Universiteit van Amsterdam
Zhang, M., & Chen, Y. (2018). Link prediction based on graph neural networks. In *NeurIPS* (pp. 5171–5181).
Zhang, S., Yao, L., Xu, X., Wang, S., & Zhu, L. (2017). Hybrid collaborative recommendation via semi-autoencoder. In *ICONIP* (pp. 185–193). Springer.
Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma W.-Y. (2016). Collaborative knowledge base embedding for recommender systems. In *KDD* (pp. 353-362). ACM.
Zhang, Y., & Ma, Q. (2020). Doccit2vec: Citation recommendation via embedding of content and structural contexts. *IEEE Access, 8,* 115865–115875.
Zhao, W., Yu, Z., & Wu, R. (2021). A citation recommendation method based on context correlation. *Intelligent Data Analysis, 25*(1), 225–243.
Zhou, X., Ding, L., Li, Z., & Wan, R. (2017). Collaborator recommendation in heterogeneous bibliographic networks using random walks. *Information Retrieval Journal, 20*(4), 317–337.
Zhou, R., Xia, D., Wan, J., & Zhang, S. (2020). An intelligent video tag recommendation method for improving video popularity in mobile computing environment. *IEEE Access, 8,* 6954–6967.
Zhuang, F., Zhang, Z., Qian, M., Shi, C., Xie, X., & He, Q. (2017). Representation learning via dual-autoencoder for recommendation. *Neural Networks, 90,* 83–89.