Semantic User Behaviour Prediction in Online News

Nina Kjekstad and Elida K. Reknes

Department of Computer Science (IDI), NTNU, Trondheim, Norway
ninakjeks@gmail.com
elida.reknes@gmail.com

Abstract. Predicting user behaviour has grown increasingly important in the context of news recommendation over the years. Knowledge of behavioural patterns offers valuable insight for developing efficient and user friendly services, ensuring good user experiences for users and increased revenues for companies. Though very useful, these user recommendations can be difficult to make and the news domain poses its own, unique challenges for the task. As a result of the news domain's volatile nature, sparse user-item matrices in excessively high dimensional space is a prominent challenge in news recommendation. This makes the utilization of traditional user modeling and recommendation methods difficult, especially due to the domain's inherent need to recommend unseen items, known as the cold start problem. Extracting abstract content descriptions by traditional topic modeling, and the more recently developed method of topic embedding, on the rich text objects provided by the news domain, new articles can be compared to older articles with more collected context and data in a meaningful and efficient way. By organizing these articles in clusters based on topic proportions, useful topic combinations can be elicited. Using these clusters to create user models based on composite interests, newly introduced articles can be recommended to users based on their topic compositions. The issue of finding a compromise between optimization of performance in subtasks and the final prediction is prominent in the experiments and results, and will be crucial in future works if the method described in this article is to be useful in a real life context. Despite this challenge, extensive experiments on two comparable real-world datasets did show improvements on user prediction results.

Keywords: News Recommendation · Topic Modeling · Community Detection.

1 Introduction

Recommender systems (RS) have been a topic of interest within research and customer services for years, proving itself an efficient and popular method of improving user experiences and revenue [9]. Certain issues have been persistent across domains, such as creating efficient and useful recommendations on sparse datasets. The sparsity of data is a natural consequence of users interacting with a small fraction of the total numbers of items, or lacking information about the items themselves.

The area of news recommendation also faces more unique challenges such as the frequency at which new items are added and the need to recommended items with little or no user history available, further amplifying the sparsity of the data. This is also reflected in the demand for item recency, the recommendation of older news articles is rarely of interest to the user. In a research field which has often relied on extensive purchase histories for both user- and item-based models, these features requires certain adaptions in order to make useful recommendations.

Explicit user feedback is also a rarity, especially measured against the total number of users and the frequency of article consumption per user [5]. Registered user interests are often based only on

NOBIDS 2018

whether a particular user has read the article or not, with no guarantee of whether the user enjoyed the article itself or a composition of the topics it encompasses. It is also important to note that a user may have enjoyed an article they have not read, and assumptions about articles that are not read are also difficult to make, especially due to the mentioned high turnover rate and volumes of articles. Additionally, certain articles, such as breaking news, are commonly popular across all groups and types of users regardless of their composite personal interests.

These features produces many variables and considerations that needs to be addressed when recommending news articles. This article provides a method that deals with these challenges by modeling topical, dimensionality reduced representations of documents. Extraction of abstract topics from text corpora enables generation of rich item information based on content at introduction time, and further allows for automated document comparison and user models built on expressed interests in composite topics, as well as alleviating the data sparsity and thus the cold item start problem. By connecting articles and users to higher level topical clusters, users may also be recommended articles in a broader range than if solely using each user's read history, facilitating exploration and avoiding monotone filter bubbles. The method shows improvements on user prediction results, but a key finding which should be emphasized is the conflict between optimizing goals for subtasks and optimal performance in the prediction task.

2 Related Work

2.1 Recommender Systems

In the context of recommender systems, community detection through clustering has proven to be an efficient tool for personalized recommendations extracted from interactions between users and rather persistent items such as books [21] or movies [19] [4] [18]. Similar to traditional collaborative filtering, recommendations are made based on knowledge gained from users' preferences generalized in user models, and users with similar user models.

However, a key component of a user model is the collected knowledge of user behaviour through user feedback (explicitly or implicitly [17]), potentially putting high demands on the user. Within the news domain, users will often read several articles each day, but likely not a significant portion of the total number of articles published. Furthermore, they are also not likely to take the time to, nor appreciate a prompt to, rate all articles they read. This can contribute to very sparse matrices when creating user models based on user-item interaction history [17]. This is expected to be especially prominent in the news domain due to volatility in articles' relevance, through high turnover rates, different types of articles with varying lifespans and the irregular nature of individuals' behaviours.

2.2 Topic Modeling

In order to deal with the sparsity issue as a consequence of the volatile news domain, a content based approach may be suitable, comparing content of the news articles and thus not mainly relying on user-item interaction. As simply comparing the articles' words will result in a high-dimensional sparse matrix, dimensionality reduction may be applied in order to achieve a more general representation of the articles' content, and consequently user history in terms of topic interests.

Topic modelling allows for this, by discovering overall topics from unstructured data. While discriminative models have been preferred for many natural language processing tasks due to their clear relation to the traditional classification task of finding the label Y when given the features X, generative models provide more flexibility and can be more suited for cases lacking fully labeled datasets or where the opportunity to generate sample data is advantageous [2] [12] [13].

A topic model which has inspired many further models is the generative statistical model Latent Dirchlet Allocation (LDA). A key feature of the Dirichlet distribution in topic modeling is that it assumes that each topic has a few high-probability words connected to them, and that each document in turn has a few high-probability topics connected to them, a natural assumption for many topics and documents which allows for each document to be presented as a composition of topics [1]. Though it has been a useful method for dimensionality reduction of documents, and has evolved through the years, the need for a pre-defined number of topics is a potential weakness in the field of news recommendation. This is also the case for nonnegative matrix factorization methods, often used in recommender systems, such as Alternating Least Squares [11]. Topics may frequently disappear or emerge, as well as defining terms of a topic changing over time, making LDA potentially non-compatible with the need to capture current and updated topics.

2.3 Representation Learning in Topic Extraction

An alternative method of topic extraction, addressing some of the these issues, is by the use of representation learning. Representation learning is a manner to extract semantic similarities and interesting relationships between entities, such as words, and has gained great attention within deep learning research. It also allows for sparsity alleviation through compressed representation, and efforts to exploit sparsity in the generation phase is common practice [13][7]. Extensions from word embeddings to topic embeddings, describing paragraphs or full documents, have mostly been tested within information retrieval, classification and sentiment analysis [8] [15]. One such effort is TopicVec, built on a generative word embedding method designed to accommodate for later use of latent factor utilization such as topic detection [13]. TopicVec capitalized on this in [12]. Simply put, documents in TopicVec are presented as "bags-of-vectors" in which the elements refer to embeddings of the words contained in the document.

Some attempts have been made to incorporate the methods in the news domain, with several clustering topic modeled news articles but never extending to recommendation [8]. Others have used word and topic embeddings for recommender systems, enriching content information through external textual data sources [15], but not to recommend textual items in themselves. Embeddings of other items such as venue checkins and product purchase sequences of item purchases or music services have also been used in order to improve recommendations [16].

2.4 Clustering Topical Communities

With a more general and comparative representations of the documents, topical communities may be detected through clustering, which has been applied to similar corpora with promising results [14] [20]. Users may then be modeled in the same dimensions as the articles, and be assigned membership to multiple communities based on the topics of the articles they have shown interest in. New articles may then be targeted to members of communities, with the topical presentation that fits with the user model eliminating the need for further interaction history. This is also seemingly realistic way to represent users, as people's interests are nuanced and most people in reality do not exclusively belong to only one community [22], as illustrated in figure 1 and 2.

In the news domain, a suitable clustering algorithm should be able to handle large datasets, many clusters of varying sizes and preferably also non-flat geometrical shapes. k-Means [6] and

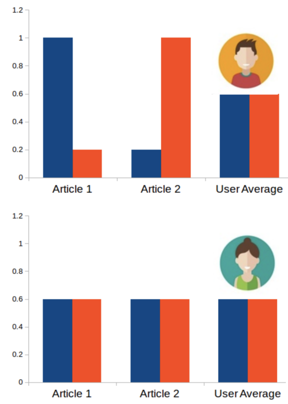

Fig. 1. User models resulting in averaging topic interests

Fig. 2. User models resulting in attachment to composite topical interests

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [3] complements each other and fulfills these requirements. Though k-Means requires a predefined number of clusters and is more suitable for clusters of even size and flat geometry, which is not optimal for the news domain, it is simple and scalable, and suitable as a well-known general-purpose baseline algorithm. In contrast, DBSCAN may handle clusters of non-convex shapes and uneven sizes. The algorithm is also scalable and does not require a predefined number of clusters, making it promising for clustering in the dynamic and composite news domain.

3 Method

As described earlier in the article, the news recommendation area struggles with providing a data foundation on which traditional recommendation methods of collecting user history on item before recommending them, or denoting items by simple categorizations, can perform optimally. The process presented in the following aims to improve recommendation of news articles by performing predictions which mainly relies on content, and for this purpose utilizes topic extraction and clustering methods to build user models and embed new articles into an existing space of set dimensions. Letting the algorithms for topic extraction and clustering be denoted by parameters γ and ζ, a set of new articles D and γ-parameters are sent into γ, outputting dimensionally recuded document vectors V, with dimensions corresponding to topics. V and ζ-parameters are used by ζ to generate a set of topical clusters C of documents. Unseen documents may then be embedded by γ, and connected to topical clusters in C. Using each user's document base D_u, each user is represented by a cluster membership vector ω_u, being input together with C, to the predictor generating recommendations η. Figure 3 shows the implementation at a conceptual level, also presented in in algorithm 1. A brief description of the main objectives and setup, further described in [16], will be outlined next.

Algorithm 1 Recommend new articles after model and cluster intialization

procedure TopicModel(D)
  for each document d in D **do**
    $d_{vector} = \gamma.d$
  end for
  γ.update()
  Return d_{vector}
end procedure

procedure AssignClusters(V)
  for each document v in V **do**
    $c_v = \zeta.minDistance(v)$
  end for
end procedure

procedure Predict(ω,V)
  for each user ω_u in ω **do**
    $Rec_u = \eta$.predict(ω_u,V)
  end for
  Return Rec
end procedure

procedure Recommend(D)
  $V = TopicModel(D)$
  AssignClusters(V)
  Rec = Predict(ω,V)
  Return Rec
end procedure

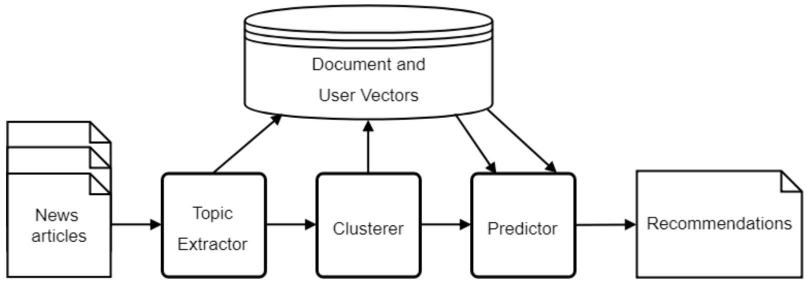

Fig. 3. Basic architecture for article recommendation.

In the context of this paper, the topic modeling will either infer the the full corpus likelihood in TopicVec:

$$p(\mathbf{Z}, \phi | \alpha, \mathbf{D}, \mathbf{A}, \mathbf{V}, \mathbf{T}). \tag{1}$$

Here, the training objective is to discover the word-topic and document-topic distributions, $\mathbf{Z}$ and ϕ, given the hyperparameters and documents, and fixing the word embedding variables $\mathbf{A}$ and $\mathbf{V}$ along with the topic candidates $\mathbf{T}$ which is inferred from the whole corpus. Extensive mathematical proof can be found in [12].

In the case of LDA, the corpus likelihood is calculated as

$$p(D \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta) \right) d\theta_d. \tag{2}$$

where N represents a word-topic combination, w is the word and z is the topic distribution of the word. M denotes the document in the collection and θ is the topic distribution of the document, with θ_m representing the distribution for the document m. Both models utilizes Dirichlet priors to capture the relationship between words topics and documents and topics.

3.1 Experimental setup

Due to lack of access to a complete dataset reflecting the news domain in terms of sufficient attributes on both articles and user history, two public datasets were mainly used, complementing each other. The *News Aggregator*[1] dataset, containing categorized headlines, was used to simulate the categorized article documents, while a data collection of user submissions at the social content aggregation and discussion website *reddit*[2] was used to cover user interactions. Thus it should be noted that only the latter was available for recommendation testing, while the first provided a comparison of the reddit data and real life news data.

Topic extraction is performed on both data sets, with each model utilizing either LDA [2] or TopicVec [12] before being clustered. LDA is run with a grid searching parameter t (number of topics), in the range $[0, 230]$ with step size 10. Each model is then evaluated on a held-out validation set by measuring perplexity, bounds and coherence, from which a representative selection of models

[1] https://www.kaggle.com/uciml/news-aggregator-dataset
[2] https://files.pushshift.io/reddit/

is chosen. When TopicVec is used, the algorithm is run on top of the continuous word embeddings provided by its predecessor PSDVec, as recommended from [12], in 500 dimensions across several choices of topic dimensions chosen to ensure comparability to the LDA runs and a large enough range to observe trends.

Extracted topics are used to represent documents as topic vectors. The topic vectors resulting from both LDA and TopicVec are then clustered by the k-means or HDBSCAN algorithm, applied to document vectors resulting from the varying numbers of topics in the previous step, and with a range of clustering parameters in order to observe trends for different numbers and sizes of clusters. Users are later connected to the resulting topical communities based on their interaction history.

The best parameters are chosen based on Silhouette coefficient and Calinski-Harabasz (CH) index, and the best clusterings are then used to create user models based on the users' interactions. Due to the inability to assert these evaluation metrics as a guideline for the best possible clusters for the recommendation task, some visualization and qualitative choices are used to pick out some complimentary models for further testing. These qualitative choices were aided by evaluating 2-D and 3-D visualizations of clusterings on topic modeled documents, also compared to clusterings resulting from the pre-defined categories and subreddits. Another qualitative consideration when selecting models, was choosing models which were not top performing in terms of metrics but provided a compliment to quantitatively chosen models in terms of topic or cluster numbers, and the ratio between the two.

The user model is built by classifying all unique documents the user either has published or commented on. The users memberships to each topical community is then decided by number of interactions within each community. More precisely membership strength, ω, is given by

$$\omega_{u,c} = \log(count_{u,c} + 1) \tag{3}$$

where $count_{u,c}$ is the number of unique document user u has interacted with in cluster c. To emphasize activity involving more rare users and clusters, inspired by [10], $\omega_{u,c}$ is supplemented with deviations of average for user u and average for cluster c from the overall average, denoted as δ_u and δ_c. The score $s_{u,i}$ is then:

$$s_{u,i} = \omega_{u,c} + \delta_u + \delta_c \tag{4}$$

, where document i belongs to cluster c, and $\omega_{u,c}$ (equation 3) is initial membership weight of user u in cluster c.

During prediction, 230 documents are attempted predicted for each user, where 30 of these are known to be in the user's interaction history. The ranked list is Mean Average Precision (MAP) evaluated at position 3, 5, and 10, in addition to the full list and 30, being the number of articles we wish to recommend. A ranked list sorted by Ranking Accuracy (RA) is also generated, appearing to better capture the quality of recommendations, considering the characteristics of each prediction evaluation metric. Documents classified as noise are not predicted. Both metrics and their dependencies are displayed in table 1.

4 Result

Table 2 summarizes the models yielding the best topical communities in terms of evaluation metrics, from each of the different experiments. In the table, t represents the number of topics, k the k-

Evaluation Metric	Equation	Parameters
Average Precision	$AP@k = \sum_{i=1}^{k} P_i \cdot \Delta R_i$	k = number of top elements
Mean Average Precision	$MAP = \frac{1}{\lvert Q \rvert} \sum_{q \in Q} AP@k$	Q = set of all users
Percentile Ranking	$\overline{rank} = \frac{\sum_{u,i} \alpha_{u,i} \cdot rank_{u,i}}{\sum_{u,i} \alpha_{u,i}}$	$\alpha_{u,i} = 1$ if user u interacted with article i, 0 otherwise
Ranking Accuracy	$RA = \frac{50\% - \overline{rank}}{50\%}$	-

Table 1. Prediction evaluation metrics [16].

parameter in k-Means, and s and c the minimum sample and cluster parameters in DBSCAN[3]. The prediction result for each of the models presented previously in table 2 is displayed in figure 4. The combination of TopicVec and DBSCAN produced such excessive numbers of outliers and very small clusters that it seemed of little use for the intended purpose, and is thus not included.

Model	**t**	**Cluster Parameters**	**Training Set**	**Silhouette**	**C-H**	**MAP@TOTAL**	**RA**
LDA1	4	NA	Reddit	NA	NA	0.35	0.53
LDA2	15	NA	Reddit Title	NA	NA	0.30	0.35
LDA+HDBSCAN1	100	s=10, c=15	Reddit Title	-0.98	6.19	0.13	-0.28
LDA+HDBSCAN2	15	s=5, c=5	Reddit	-0.64	1.01	0.14	-0.18
LDA+KMEANS1	4	k=4, distance=euclidian	Reddit	-0.01	2.05	0.22	0.34
LDA+KMEANS2	15	k=4, distance=KL	Reddit Title	-0.01	2.21	0.14	0.00
LDA+KMEANS3	100	k=5, distance=euclidian	Reddit Title	-0.03	2.58	0.12	-0.28
LDA+KMEANS4	5	k=10, distance=euclidian	Reddit	-0.02	2.01	0.30	0.46
TV	5	NA	Reddit	NA	NA	0.38	0.52
TV+KMEANS1	5	k=3	Reddit	0.87	231,858.39	0.13	0.64
TV+KMEANS2	5	k=4	Reddit	0.83	231,858.39	0.20	0.22
ORIGINAL	15	NA	Reddit Title	NA	NA	0.15	0.01
RANDOM	15	NA	Reddit Title	NA	NA	0.15	0.00

Table 2. Description of compared models displayed in figure 4.

Different combinations of parameters provides prediction results with very different qualitative characteristics, but comparing to predictions using original labels and random recommendations, the models obtain better MAP and RA than both. With models using both topic modeling and clustering, RA shows improvements on random, ranged from 34%-64%, for models which performed

[3] Using Hierarchical DBSCAN implementation (https://hdbscan.readthedocs.io/en/latest/parameter_selection.html)

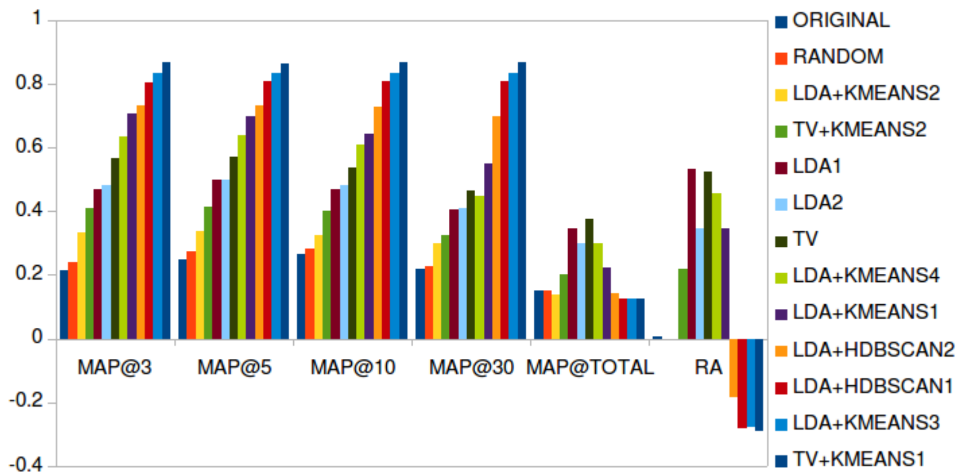

Fig. 4. Prediction results.

well. Other models gains improvements ranging from 18%-278 % on top 3, 5, 10 and 30 positions for MAP, but declines in performance on RA by $18\% - 28\%$.

As can be observed, several models perform notably better when only evaluating the upper part of the ranked list (top 3, 4, 10, and 30), but have an excessive drop in performance when evaluating the full list. In Figure 5, the distribution of positive hits are highlighted, showing some interesting tendencies. The models which obtains the higher MAP-values, have a very high focus of positive articles in the very first part of the list and in compact groups between the 50th and 75th percentile in the lower half of the recommended list. Those with lower MAP-values, generally obtain higher RA-scores. As seen in Figure 5, these models shows more evenly distributed positive hits in the list, and have a heavier focus in the first half percentiles of the recommendations, but suffers in MAP evaluation due to the gap present between the very first hits and the 10-25th percentile. Combining topic modeling and community detection, appears to shift a large bulk of the positive hits 25-50 percentiles lower down in the list. While this proves to increase MAP-scores, it appears that the concentration of positive hits in such high percentiles could indicate that the RA provides a more appropriate evaluation criteria.

An important and interesting finding, is that the models chosen for their promising evaluation scores in some ways perform worse than those chosen for comparison reasons or by qualitative consideration. Two such qualitatively chosen models are LDA+KMEANS4 and TV+KMEANS2, performing well within RA and thus presenting more of the relevant documents in the earlier part of the recommendation list. When using LDA, selections of lower-quality clusterings sometimes yielded better prediction results, highlighting that clustering may prioritize qualities which are not appropriate or optimal for prediction. This was also observed when using topic embedding, where some models with qualitatively chosen embeddings combined with lower-performing clusterings yielded prediction results that are preferable in a news recommendation case, compared to models with more optimal clustering evaluation metrics.

Consequently, one of the main challenges throughout the experiments has been optimizing unsupervised training models across different sub-goals to meet the prediction end goal. As observed,

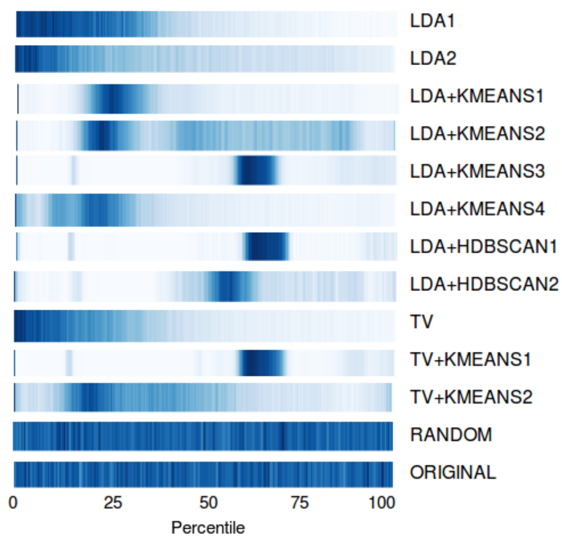

Fig. 5. Heat maps for where in the list the expected documents are positioned in the ranking. Darker color indicates higher occurence of expected documents at this position.

eliciting the best topical clusters for the explicit purpose of user recommendation is challenging as different evaluation metrics may be conflicting, and in turn, the learning objectives for topic extraction may not provide the optimal results for the purpose of clustering either. This especially relevant for topic embedding which is traditionally tested on tasks such as classification. The alignment of goals across technologies in the different steps may be complicated, and discerning the optimal solutions for the next step while relying on unsupervised training can be difficult.

5 Conclusion and Further Work

This article addresses the sparsity issue in the new domain by utilizing dimensionality reduction through semantic analysis and clustering. In order to overcome the lack of sufficient user feedback and provide better recommendations, this is implemented in a pipeline conducting document embedding, extraction of high level topical clusters, user modeling, and prediction.

A key motivation for the model was to provide set-length topical representation of documents and users, which are possible to generate the instance a new document is introduced to the system. In doing this, we have provided a model which is capable of recommending previously unseen documents, a vital property for a news recommender system. All combinations of topic extraction and clustering does perform better than random and the use of original labels to recommend, both in terms of MAP and RA.

Though the system does support the recommendation of previously unseen documents through topic modeling and clustering, aligning the subgoals of optimal topic representation and clustering proved difficult. The quantitative evaluation metrics used to evaluate clusterings did not necessarily make for better recommendation models and the evaluation metrics for the recommendations themselves also proved to be misleading at times.

An important note is the noisiness and partially unrealistic comparison to the news field the tested dataset produces. A more complete and directly news related dataset with the possibility of connecting a single users history over time is necessary to produce more realistic testing enviroments.

The model described in this paper, does not consider the effect of long-term and short-term interests among users, nor the potential differences in life span for articles of different types and topics. Adding considerations for these time sensitive attributes could provide a model which not only gives better recommendations, but which remains able to give valuable predictions over time.

Though these considerations and additions to the described model could further improve recommendations, a main challenge remains the alignment of subgoals in the tasks of topic modeling and clustering and the final recommendations. In order for the methods to be useful, a way to automatically discover optimal variables across the process is important, and new evaluation metrics may need to be utilized or created for the task.

Acknowledgement

The authors would like to thank L. Zhang and P. Liu at the Department of Computer Science at NTNU for their support.

References

1. Blei, D.M.: Probabilistic topic models. Commun. ACM **55**(4), 77–84 (Apr 2012). https://doi.org/10.1145/2133806.2133826, http://doi.acm.org/10.1145/2133806.2133826
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (Mar 2003), http://dl.acm.org/citation.cfm?id=944919.944937
3. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. vol. 96, pp. 226–231 (1996)
4. Feng, H., Tian, J., Wang, H.J., Li, M.: Personalized recommendations based on time-weighted overlapping community detection. Information & Management **52**(7), 789–800 (2015)
5. Gulla, J.A., Zhang, L., Liu, P., Özgöbek, Ö., Su, X.: The adressa dataset for news recommendation. In: Proceedings of the International Conference on Web Intelligence. pp. 1042–1048. ACM (2017)
6. Hartigan, J.A.: Clustering algorithms (1975)
7. He, J., Hu, Z., Berg-Kirkpatrick, T., Huang, Y., Xing, E.P.: Efficient correlated topic modeling with topic embedding. CoRR **abs/1707.00206** (2017), http://arxiv.org/abs/1707.00206
8. Hui, S., Dechao, Z.: A weighted topical document embedding based clustering method for news text. In: 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference. pp. 1060–1065 (May 2016). https://doi.org/10.1109/ITNEC.2016.7560526
9. Isinkaye, F., Folajimi, Y., Ojokoh, B.: Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal **16**(3), 261 – 273 (2015). https://doi.org/https://doi.org/10.1016/j.eij.2015.06.005, http://www.sciencedirect.com/science/article/pii/S1110866515000341
10. Koren, Y.: Collaborative filtering with temporal dynamics. Commun. ACM **53**(4), 89–97 (Apr 2010). https://doi.org/10.1145/1721654.1721677, http://doi.acm.org/10.1145/1721654.1721677
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8) (2009)
12. Li, S., Chua, T., Zhu, J., Miao, C.: Generative topic embedding: a continuous representation of documents (extended version with proofs). CoRR **abs/1606.02979** (2016), http://arxiv.org/abs/1606.02979

13. Li, S., Zhu, J., Miao, C.: A generative word embedding model and its low rank positive semidefinite solution. CoRR **abs/1508.03826** (2015), http://arxiv.org/abs/1508.03826
14. Li, X., Yan, J., Fan, W., Liu, N., Yan, S., Chen, Z.: An online blog reading system by topic clustering and personalized ranking. ACM Transactions on Internet Technology (TOIT) **9**(3), 9 (2009)
15. Musto, C., Semeraro, G., de Gemmis, M., Lops, P.: Learning word embeddings from wikipedia for content-based recommender systems. In: Ferro, N., Crestani, F., Moens, M.F., Mothe, Josiane Silvello, G. (eds.) Advances in Information Retrieval. pp. 729–734. Springer International Publishing, Cham (2016)
16. Nina, K., Elida, R.: Semantic User Behaviour Prediction in Online News. Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway (2018)
17. Oard, D.W., Kim, J.: Implicit feedback for recommendation systems (1998)
18. Shani, G., Meisles, A., Gleyzer, Y., Rokach, L., Ben-Shimon, D.: A stereotypes-based hybrid recommender system for media items (2007)
19. Wen, Y., Liu, Y., Zhang, Z.J., Xiong, F., Cao, W.: Compare two community-based personalized information recommendation algorithms. Physica A: Statistical Mechanics and its Applications **398**, 199–209 (2014)
20. Wu, Y., Ding, Y., Wang, X., Xu, J.: A comparative study of topic models for topic clustering of chinese web news. In: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. vol. 5, pp. 236–240. IEEE (2010)
21. Xin, L., Haihong, E., Song, J., Song, M., Tong, J.: Book recommendation based on community detection. In: Joint International Conference on Pervasive Computing and the Networked World. pp. 364–373. Springer (2013)
22. Zhang, K., Zhang, Z., Bian, K., Xu, J., Gao, J.: A personalized next-song recommendation system using community detection and markov model. In: Data Science in Cyberspace (DSC), 2017 IEEE Second International Conference on. pp. 118–123. IEEE (2017)