Tell Me What You Want: Embedding Narratives for Movie Recommendations

Lukas Eberhard
Graz University of Technology
Graz, Austria
lukas.eberhard@tugraz.at

Simon Walk
Detego GmbH
Graz, Austria
s.walk@detego.com

Denis Helic
Graz University of Technology
Graz, Austria
dhelic@tugraz.at

ABSTRACT

Recommender systems are efficient *exploration* tools providing their users with valuable suggestions about items, such as products or movies. However, in scenarios where users have more specific ideas about what they are looking for (e.g., they provide describing narratives, such as *"Movies with minimal story, but incredible atmosphere, such as* No Country for Old Men*"*), traditional recommender systems struggle to provide relevant suggestions. In this paper, we study this problem by investigating a large collection of such narratives from the movie domain. We start by empirically analyzing a dataset containing free-text narratives representing movie suggestion requests from reddit users as well as community suggestions to those requests. We find that community suggestions are frequently more diverse than requests, making a recommendation task a challenging one. In a prediction experiment, we use embedding algorithms to assess the importance of request features including movie descriptions, genres, and plot keywords, by computing recommendations. Our findings suggest that, in our dataset, positive movies and keywords have the strongest, whereas negative movie features have the weakest predictive power. We strongly believe that our new insights into narratives for recommender systems represent an important stepping stone towards novel applications, such as interactive recommender applications.

CCS CONCEPTS

• **Information systems → Recommender systems**; *Users and interactive retrieval*; *Personalization.*

KEYWORDS

Narrative-driven recommendations; Recommender systems; Empirical analysis

ACM Reference Format:
Lukas Eberhard, Simon Walk, and Denis Helic. 2020. Tell Me What You Want: Embedding Narratives for Movie Recommendations. In *31st ACM Conference on Hypertext and Social Media (HT '20), July 13–15, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3372923.3404818

HT '20, July 13–15, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7098-1/20/07…$15.00
https://doi.org/10.1145/3372923.3404818

SUBMISSION

"**[Request] Movies about writing/writers.**
Two of my favourites are *Secret Window* and *Stranger Than Fiction.* I also liked *The Ghost Writer.* [...] I'm not a fan of horror. I know there are probably a lot of 'inspirational' movies about writing out there (I vaguely recall one with Sean Connery?). [...]"

COMMENTS

"Adaptation."

"Sean Connery movie was Finding Forrester."

⋮

Box 1: *Request* & *Suggestions* *Example.* In the request of this reddit submission[1] crowdworkers annotated three positive movies (i.e., *Secret Window*, *Stranger Than Fiction*, *The Ghost Writer*), a negative genre (i.e., *horror*), several positive keywords (i.e., *writing*, *writers*, *inspirational*), and a positive actor (i.e., *Sean Connery*). As suggestions from the reddit community, the crowdworkers extracted the movies *Adaptation* and *Finding Forrester* from the comments section.

1 INTRODUCTION

Search engines are omni-present tools designed to help users retrieve information when they specifically *know what they are looking for* (i.e., they can articulate what they want with a few simple keywords). On the other hand, users rely on recommender systems when they *are unable to specifically state what they seek* (i.e., they vaguely know what they want but can not articulate it). In that case, recommender systems allow users to *explore* large collections of items and find interesting items by, for example, browsing [10].

Problem. In contrast to those two information seeking situations, in *narrative* scenarios users have a more specific idea what they are looking for, but the information need is often too complex to be articulated in the form of a few simple keywords. For example, in online boards they describe what they are looking for in the form of a narrative request while other users provide relevant recommendations (cf. Box 1).

Results of our recent work on such recommendation scenarios [5] indicate that the problem of narrative-driven recommendations is hard and difficult to address with traditional recommender approaches. However, our research community still lacks a deeper understanding of the potential causes for the hardness of this problem. For instances, we miss insights into user preferences in narratives, and, particularly, whether users tend to illustrate their needs

[1] https://www.reddit.com/r/MovieSuggestions/comments/ssuhu

through (i) examples, (ii) by describing the characteristics of desired items, or (iii) by a specific combination of both examples and characteristics. Also, the question whether positively associated aspects (e.g., *writing*, *writers*, *Secret Window* and *Stranger Than Fiction* in Box 1) are more important for calculating recommendations than negatively associated aspects (e.g., *horror* in Box 1) is still unanswered in the recent research on this topic.

This Work. In this paper, we set out to learn more about narratives by analyzing a movie suggestion board from reddit (r/MovieSuggestions[2]). We start our study by empirically analyzing a dataset that consists of narrative requests and corresponding movie suggestions from the reddit community. To quantify the difficulty of this problem we analyze the diversity of requests and their corresponding suggestions. Further, we evaluate the effects of positive vs. negative features of users on reddit. Next, we utilize document and graph embedding techniques to (i) compute algorithmic recommendations and to (ii) evaluate the importance of features extracted from requests by comparing algorithmic recommendations with movie suggestions from the reddit community. The results of our empirical analysis reveal that community suggestions are oftentimes more diverse than requests, meaning that similar requests are frequently answered with highly diverse movie suggestions by the reddit community. The results of our recommendation experiment show that positive movies and keywords have the strongest, whereas negative movie features the weakest predictive power for narrative-driven movie recommendations on reddit.

Related Work. A special context-aware recommendation scenario called narrative-driven recommendation was proposed by Bogers and Koolen [4]. In the recommendation calculation process, besides the user history a narrative explanation of the current recommendation needs of the respective user is utilized.

Although Glenski and Weninger [6] showed that simple models are able to predict user interactions, such as likes, votes, clicks, and views, on reddit, recommending movies based on narrative requests on reddit is hard [5]. In our previous work [5], we determined the suitability of well-established recommender algorithms for calculating narrative-driven movie recommendations. For evaluation, we built a crowdsourced dataset from reddit submissions providing narrative movie recommendation requests and comments including corresponding movie suggestions by the reddit community. The obtained results reveal that the problem of predicting narrative-driven recommendations is a hard problem and needs further investigation to get a better understanding of narrative aspects.

Contrarily, in this work we use this crowdsourced dataset and follow up by introducing the first in-depth empirical analysis of movie recommendation requests on reddit. Moreover, we evaluate the embedding of narratives through the document and the graph embedding techniques `doc2vec` and `node2vec`.

In the large and well-investigated research field of recommender systems and algorithms [1, 4, 7] there exists a vast variety of studies based on `word2vec` [12] and its extensions `doc2vec` [11] and `node2vec` [8] partly exhibiting outstanding performances [2, 5, 13, 14]. Kallumadi and Hsu [9] evaluated the effectiveness of query-based interactive movie recommendations on IMDb data using graph-level embeddings. They created meta paths with different entities (e.g., users, movies, genres) to build movie networks as basis for their embeddings and obtained good results with `node2vec`.

In contrast to their work, we use multiple networks, each of which consists of nodes from the same type. Further, we combine `node2vec` with `doc2vec` embedding vectors based on textual movie information.

[2]https://www.reddit.com/r/MovieSuggestions

2 EMPIRICAL ANALYSIS OF NARRATIVES

We empirically analyze the publicly available crowdsourced dataset[3] that we extracted from reddit [3] in our previous work [5]. The dataset contains about 1,500 narrative requests that all received at least ten suggestions and about 21,000 suggestion lists including more than 43,000 individual suggestions. We list further details of our dataset in Table 1. Each request in our dataset includes one or more positive movies, which are examples of movies that the user liked before. Moreover, requests frequently include additional descriptions, such as negative movies (movies that the user did not like before), positive and negative keywords describing further aspects of the movies, positive and negative genres, and finally positive and negative examples of movie actors. In Box 1 we show a typical example of such a request from our dataset, in which crowdworkers annotated three positive movies, one negative genre, three positive keywords, and one positive actor. We also show two examples of the suggestion lists, each of them having a single suggestion.

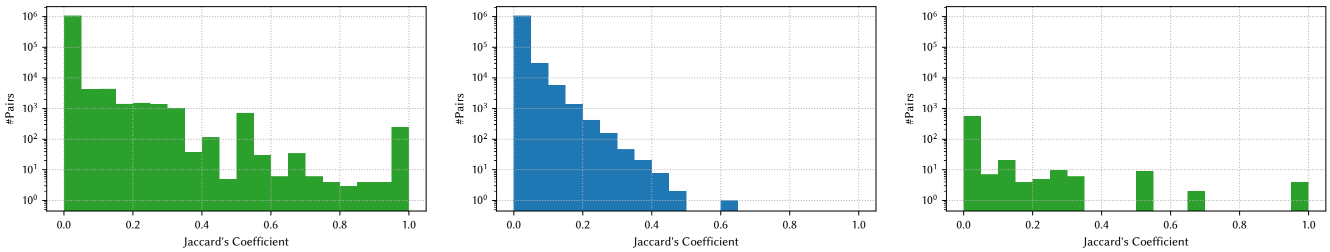
2.1 Dataset Characterization

Popularity Bias. We observe a heterogeneous distributions of positive movies and suggestions. Particularly, while the majority of the movies is mentioned only a few times, there also exist a few highly popular movies indicating a strong popularity bias. To check whether movies used as examples and suggestions are correlated we

[3]https://www.rbz.io/datasets

Table 1: *Reddit Dataset Characteristics.* This table shows the statistics of the crowdsourced reddit dataset [5].

#Requests	1,480
#Request Authors	1,244
#Movies in Requests	5,521
#Requests With Positive Movies	1,480
#Requests With Negative Movies	77
#Keywords in Requests (Without Common Words)	4,492 (3,947)
#Requests With Positive Keywords	1,202
#Requests With Negative Keywords	152
#Genres in Requests	762
#Requests With Positive Genres	459
#Requests With Negative Genres	55
#Actors in Requests	100
#Requests With Positive Actors	73
#Requests With Negative Actors	7
#Suggestions	43,402
#Suggestion Authors	7,431
Average #Suggestions per Request	29.33
Average Duration Between Request and Suggestion	31 h 41 min



(a) *Positive Movies*

(b) *Suggestions*

(c) *Positive Keywords (Pos. Movies Sim. ≥ 0.5)*

Figure 1: *Overlap of Movies & Keywords in Requests & Suggestions.* These figures (y axes on the log scale) show the Jaccard's coefficient of positive movies (a), suggestions (b), and positive keywords (c), with positive movies similarity ≥ 0.5 for all request pairs. We observe positive movies and positive keywords in requests having a longer tail with a higher probability for high similarity values as compared to suggestions. For example, we find 1043 pairs consisting of 622 (42.03%) requests with positive movies, 708 pairs consisting of 410 (34.11%) requests with positive keywords, and only a single suggestion pair with Jaccard's coefficients ≥ 0.5. These findings indicate a strong diversity of suggestions in our dataset.

compute Spearman's coefficient. We find that popular movies, such as *Fight Club* or *Memento*, are frequently used as positive examples as well as suggestions (Spearman's coefficient ρ=0.638, p<.001). In contrast to positive movies and suggestions, we do not observe any conclusive patterns in the distribution of negative movies due to their infrequent occurrences.

Hence, our initial findings indicate that automatic recommender algorithms trained on this or similar data may be strongly influenced by the popularity bias and should follow advanced strategies to account for this bias in order to, for example, increase beyond-accuracy metrics, such as diversity or novelty.

Users Prefer Dark Over Hollywood Movies. Looking at relative occurrence frequency among positive and negative keywords we find that users on reddit prefer to watch *dark* movies in *psychological* settings rather than *hollywood* or *gory zombie* movies.

Summary. Our initial dataset characterization suggests a strong popularity bias towards movies popular in this particular reddit community. The fact that there is a significant correlation between positive movies and suggestions also indicates that users frequently describe similar requests with differing movie examples. This highlights the importance of context in such narrative-driven recommendation requests as we observe no distinguishing use of positive and negative keywords. Thus, users signal their recommendation needs through specific and distinctive combinations of various features including example movies, genres, or keywords.

2.2 Suggestions Diversity

We continue our empirical analysis by investigating how diverse community suggestions are. Particularly, as reddit submissions resemble a typical discussion board structure, we expect that each following user suggesting movies recommends movies different to the already suggested ones. Moreover, as users can browse the submission lists and corresponding suggestions from the past, we also expect a high suggestion diversity across different requests. Taken all together, we hypothesize that highly diverse community responses will render narrative-driven recommendations as a challenging task for automatic recommender algorithms.

Overlapping Examples Result in Differing Suggestions. In the first step, we investigate how overlap in positive movie examples across requests relates to overlap (or lack thereof) in suggestions. Thus, we compute a standard overlap measure, the Jaccard's coefficient, of positive movies from all pairs of requests and corresponding suggestions. We plot all overlap distributions in Figure 1. We find that both requests and suggestions are typically dissimilar to the majority of other requests with average of 0.003 (sample standard deviation s=0.028) and suggestions with average of 0.008 (s=0.019). However, while there is still a substantial number of requests highly similar to at least one other request (e.g., with similarities ≥ 0.5). In contrast to requests, we do not observe such similarities in pairs of suggestions.

Varying Tastes Lead to Suggestion Diversity. One possible explanation for the diversity in suggestions may be that suggestions are provided by different users with diverging movie preferences. As we expect that users will follow their own tastes and preferences while suggesting movies, the varying user preferences will result in different suggestions even to identical requests.

To test this intuition, we start by removing all suggestion lists from deleted users (3% of suggestion lists), who are denoted as "[deleted]" in our dataset. Next, we analyze the overlap of suggesting users for requests with identical positive movies. In 230 of such request pairs we find an average overlap (as measured with the Jaccard's coefficient) between suggesting users to be 0.010 (s=0.024). Similarly, the average overlap of suggesting users for the request pairs with similarities ≥ 0.5 is as low as 0.008 (s=0.020). Thus, this observation corroborates the previous finding suggesting that typical users respond sporadically by answering only to a small number of posted requests or by providing only a small number of suggestions. Finally, to assess the effect of user personal preferences on the suggestions that they make we first eliminate the effects of confounding factors, such as additional positive keywords or genres. Therefore, we control for these features and extract a subset of requests with identical positive movies and no additional information. We find 14 such pairs comprising 17 requests (1% of all requests). Among those request pairs we find no overlap between responding users. The average similarity of 0.081 (s=0.050) for suggestions between these pairs aligns with our hypothesis about the importance of user preferences when providing suggestions. However, we retain from making any conclusions based on these results due to a small number of such request pairs in our dataset.

Community Responses Take Long Time. Further, the time span between the posting of requests and the suggestions from the community is another possible factor leading to diversity in responses. We find that for the request pairs with positive movie similarities ≥ 0.5, the average time span between the requests is about one year and eight months with sample standard deviation being about one year and four months (we obtain comparable results for similarity threshold of 1). As numerous new movies are released in such prolonged periods of time, newer suggestions possibly also include newly released movies. This result indicates that the community responses tend to be up-to-date and that the recency may be an important factor when providing narrative-driven recommendations.

2.3 Differentiating Between Requests

Our findings so far confirm our intuition that community responses are highly diverse, at least with respect to highly overlapping positive movie examples. This constitutes a further evidence hinting towards the importance of the context provided by a given narrative as well as the importance of the additional information that users provide in their requests. In the next step of our empirical analysis, we investigate such additional information in more detail. To that end, we continue by analyzing positive keywords in requests.

Keywords Differentiate Between Similar Requests. In analogy to positive movies we compute Jaccard's coefficient between positive keywords for all pairs of requests. Similarly to positive movies and suggestions, we find that, on average, positive keywords are highly dissimilar to each other with average of 0.004 (s=0.031). To further investigate whether the positive keywords are able to discriminate between requests with highly similar positive movies (similarities ≥ 0.5), we look more closely on the distribution of the keyword similarities (cf. Figure 1c) for this subset. From those pairs we extract 619 pairs with both requests including positive keywords (457 requests in total). The average keyword similarity for this selection of request pairs is 0.031 (s=0.118).

While these results indicate that keywords play an important role in distinguishing the user request between otherwise highly similar requests, they may be distorted by additional information, such as negative movies, negative keywords, positive and negative genres, or positive and negative actors. Thus, we once more extract a subset of requests with highly similar positive movies (i.e., with similarities ≥ 0.5) but that include only positive keywords as the additional information. This leaves us with 245 request pairs comprising 224 such requests. We find a similar distribution of keyword similarities for those requests as in Figure 1c. The average of this distribution is 0.032 (s=0.127), thus corroborating our previous results on the distinctive role of the keywords in recommendation requests. Moreover, we find that the average similarity of the suggestions corresponding to this request subset is low at 0.061 (s=0.067), hinting once more at differentiation potential of positive keywords. Note that we repeat the same calculations for positive movie similarities of 1 and obtain comparable results.

Thus, our analysis suggests that users frequently use overlapping positive movies but differing keywords to further refine their requests, which highlights the importance of keywords for the community when suggesting movies. Based on our findings, we argue that the problem of narrative-driven movie recommendations is not as simple as "filtering based on mentioned movies" but requires the inclusion of further key aspects from requests, such as keywords.

3 FEATURE IMPORTANCE

In this chapter, we complement our empirical analysis with evaluation of feature importance within a practical application of narrative requests from reddit in a recommender system. To train our models we use the publicly available IMDb dataset and supplement it with user reviews we collect for all movies on IMDb. Note that we use `doc2vec` for movie descriptions and user reviews, while we construct graphs based on overlaps of characteristics between movies (e.g., common genres, plot keywords, or user ratings) to create vector embeddings using `node2vec`. Finally, to generate recommendations, we calculate similarity between the embedding vectors of all movies and the embeddings of the features provided in the reddit requests and select the most similar ones. In a preprocessing step, we remove all stopwords from movie descriptions and user reviews. Further, we only keep movies with at least 1,000 user ratings, at least one user review, a movie description, and at least one person in the cast, to account for the data sparsity.

Note that we scale edge weights and filter edges via thresholds for some of the resulting graphs (cf. individual paragraphs) to account for noise from rarely occurring movie characteristics or to smooth the weights by downscaling.

3.1 Source & Training of Embeddings

`doc2vec`. We use textual data about movies collected from IMDb for all movies in the dataset. We leverage their descriptions containing plot summaries, synopses as well as user reviews to train the `doc2vec` embeddings.

`node2vec`. We use data from IMDb to create movie graphs for embedding through `node2vec`. Particularly, we create several movie graphs based on user ratings, genres, plot keywords, casts and crews, and years, each of which represents a separate `node2vec` model.

Model Training & Hyperparamter Optimization. To train and optimize our models we use both the reddit dataset and the IMDb dataset. First, we create a validation and a test set by chronologically splitting the reddit dataset (80%/20%). Second, we train our embeddings on IMDb data and use the reddit data from the validation set to execute an exhaustive grid search experiment for optimizing hyperparameters for the `doc2vec` and `node2vec` models.

3.2 Calculating Movie Recommendations

After our models are trained, we calculate similarity between movie embeddings and the requests. Specifically, we use request features for our recommendation calculations as follows.

Movies. For each `doc2vec` and `node2vec` model, we compute the cosine similarity of the embedding vector of a given positive movie to all other movie embeddings from the same model. In case that multiple positive movies are given in one request, we sum up the computed cosine similarities for each movie in our dataset, and subtract similarities for all given negative movies.

Genres. To include genres in the computation of recommendations we first represent each genre from a given request by its corresponding word vector from the doc2vec model. We add up all word vectors for positive genres and subtract all vectors for negative genres, if provided. Finally, we compute the cosine similarities between the resulting vector and all movie vectors from the doc2vec model.

Keywords. We use all positive keywords from requests as input for our doc2vec model to obtain a vector representing positive keywords. If negative keywords are provided, we use them to obtain another vector representing negative keywords, and subtract it from the positive keywords vector. Finally, we compute the cosine similarities from the resulting vector to all movie embeddings from the doc2vec model.

Actors. If positive actors are mentioned in a request we treat them as keywords and add their names to all other keywords to be inferred in our doc2vec model. If negative actors are given we remove all movies with these actors in the cast, comparable to our previous work [5].

Generating & Evaluating Recommendations. Before we generate recommendations we first filter out predecessors and successors for movies specified in the request, as we assume that users do not want to receive a list of movie series they are already familiar with [5]. Further, we consider the time (i.e., the year) a request was created and remove movies that were released later. After these filtering steps, we then evaluate the importance of positive as compared to negative mentioned features in the requests as well as the importance of all features individually in isolation of other features. Finally, we linearly combine cosine similarities of individual models into a single similarity measure for each movie from our dataset. To evaluate the importance of features we treat the coefficients of the linear combination of our models as parameters that we once more optimize via exhaustive grid search.

In each experiment, we evaluate the performance of our recommendation approach on the test set. For evaluation, we limit our final recommendation lists to ten movies[4] and compute precision, recall, and F1 score [5, 15] by comparing our recommendations to the suggestions of the reddit community. As baselines we utilize a most popular approach and use the results obtained in our previous study [5]. Note that the target of this paper is not to beat every existing state-of-the-art approach tackling the narrative-driven recommendation problem but rather to analyze the importance of user preferences and request features.

3.3 Results

Positive vs. Negative Information. To measure the impact of explicitly mentioned negative features, we first conduct our experiments leveraging only positive information including negative information only in the second iteration of our experiments. Our results reveal that additionally considering negative features in requests in most cases leads to slightly worse results than with positive features only.

[4]Recall@10 and F1 score@10 have mean upper limits of 0.45 and 0.60 respectively, as the average number of movie suggestions from the community per request is 29.22 in the test set.

Individual Features. We obtain the highest F1 score of 0.086 with the doc2vec model followed by the graph-based node2vec models with user ratings (0.039) and plot keywords (0.034), and the doc2vec model with inferred keywords from requests (0.034). One possible reason for the increased importance of movie descriptions and user reviews might be that the doc2vec model is trained on a large amount of textual information, potentially capturing additional latent attributes that are missing in the ratings and plot keywords that are used to train the respective node2vec models. Note that all node2vec models only use the given movies as input without considering further information provided in a request, such as keywords or genres.

In summary, our results with individual features indicate that narrative-driven movie recommendations exhibit only minimal overlaps with human suggestions. The most important feature individually are positive movies as the result of the doc2vec model shows. Additionally, the importance of keywords is in line with the results of our empirical analysis, where we found out that keywords are important to discern between otherwise similar requests.

Feature Combinations. We find that movie descriptions and user reviews are most important with a weight of 0.9, followed by keywords in requests with a weight of 0.8. All other models exhibit comparably low weights, with the years and casts and crews models being completely ignored with weights of 0. Using this weighted model combination we obtain an overall F1 score@10 of 0.115.

In summary, the results show that our approach (F1 score@10 of 0.115) outperforms the most popular baseline (F1 score@10 of 0.039) and is comparable with the state-of-the-art approaches from our previous work [5]. It almost reaches the performance of their best post-filtering approach using doc2vec (F1 score@10 of 0.126).

Empirical Results vs. Prediction Results. In our prediction experiments the features with the highest weights are the movie feature based on movie descriptions and user reviews, and the keywords in requests. This finding corroborates the results of our empirical analysis, suggesting that the most important narrative aspects for recommendations are positive movies combined with keywords further describing the user needs. Moreover, a high importance of the user review feature is a further strong evidence for the popularity bias as we expect that movies popular on reddit also collect more reviews on IMDb.

4 CONCLUSIONS & FUTURE WORK

In this paper, we evaluated the importance of different features extracted from requests for movie recommendations and analyzed the effects of positive vs. negative examples. We conducted an empirical analysis of a crowdsourced reddit dataset consisting of movie recommendation requests in the form of narratives and movie suggestions in the form of comments by the community. We found that users mainly focus their requests on positive aspects, such as movies they liked or keywords that describe the movies they would like to see. For future work, we plan to supplement existing approaches by extracting (e.g., by using natural language processing techniques) and incorporating information from reddit requests already in the training phase of the embeddings.

REFERENCES

[1] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)* 23, 1 (2005), 103–145.

[2] Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[3] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The Pushshift Reddit Dataset. *arXiv preprint arXiv:2001.08435* (2020).

[4] Toine Bogers and Marijn Koolen. 2017. Defining and Supporting Narrative-driven Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 238–242.

[5] Lukas Eberhard, Simon Walk, Lisa Posch, and Denis Helic. 2019. Evaluating Narrative-driven Movie Recommendations on Reddit. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19)*. ACM, New York, NY, USA, 1–11. https://doi.org/10.1145/3301275.3302287

[6] Maria Glenski and Tim Weninger. 2017. Predicting user-interactions on reddit. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. 609–612.

[7] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1809–1818.

[8] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[9] Surya Kallumadi and William H Hsu. 2018. Interactive Recommendations by Combining User-Item Preferences with Linked Open Data. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*. 121–125.

[10] Daniel Lamprecht, Florian Geigl, Tomas Karas, Simon Walk, Denis Helic, and Markus Strohmaier. 2015. Improving Recommender System Navigability Through Diversification: A Case Study of IMDb. In *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business (i-KNOW '15)*. ACM, New York, NY, USA, Article 21, 8 pages.

[11] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. 1188–1196.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[13] Cataldo Musto, Giovanni Semeraro, Marco De Gemmis, and Pasquale Lops. 2015. Word Embedding Techniques for Content-based Recommender Systems: An Empirical Evaluation.. In *RecSys Posters*.

[14] Makbule Gulcin Ozsoy. 2016. From word embeddings to item recommendation. *arXiv preprint arXiv:1601.01356* (2016).

[15] David Martin Powers. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (2011).