A Supervised Machine Learning Approach for Supporting Editorial Article Selection

BILAL MAHMOOD, MediaFutures, University of Bergen, Norway
MEHDI ELAHI, MediaFutures, University of Bergen, Norway
FARHAD VADIEE, MediaFutures, University of Bergen, Norway
SAMIA TOUILEB, MediaFutures, University of Bergen, Norway
LUBOS STESKAL, TV 2, Norway

ACM Reference Format:
Bilal Mahmood, Mehdi Elahi, Farhad Vadiee, Samia Touileb, and Lubos Steskal. 2024. A Supervised Machine Learning Approach for Supporting Editorial Article Selection. 1, 1 (November 2024), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 ABSTRACT

Editors on news platforms play a crucial role in various editorial tasks and responsibilities. One of the key tasks carried out by editors regularly is reviewing the latest news articles and *manually* selecting a set of related articles that could be interesting for readers to explore further. While this task is important, it can pose challenges, as it may take a substantial amount of time to search the database of published articles, check their content, and hand-select the most relevant ones.

In this paper, we address this challenge by proposing an *automatic* approach that can support editors in this process and assist them in selecting related articles for a given target article. The approach is based on Supervised Machine Learning (SML) and leverages state-of-the-art text embedding models to create representations of news articles. A machine learning classifier is built using these embeddings and is utilized to predict scores for available articles based on their relatedness to a target article. The top articles are then recommended to the editor for consideration in the list of the most related articles.

We evaluated our approach using a real-world dataset received from one of the largest editor-managed commercial media houses in Norway, i.e., TV 2. The dataset includes editors' feedback on how news articles are related and has been used as ground truth to assess the effectiveness of our proposed approach. The results are promising, reflecting the effectiveness of the proposed approach in handling this task in the editorial process in the news domain.

2 INTRODUCTION

One of the grand challenges in digital environments is the growing number of news articles published online every day. It is estimated that hundreds of thousands of news articles are published globally each day by different news publishers [17]. With such a large number of articles, it is becoming increasingly difficult for online users to find relevant news articles to read.

Recommender Systems (RSs) are digital tools designed to help users find the most relevant news articles based on their interests. These systems typically analyze data collected from users while they browse news articles online, building a reading profile that represents their news preferences and affinities. These profiles are then utilized to match news articles with the users' preferences and recommend them to browse further a list of the most interesting articles. This has made news recommendation becoming central for users to find and interact with news outlets [18].

Although this process may seem straightforward, it often fails to take into account the editorial mission, which plays a significant role in the management of news publishing operations, e.g., curating, fact-checking, and selecting important news for readers to consume. Such a role ensures timely and accurate reporting of the latest news. Moreover, this involves different stakeholders and is a key part of media organizations, e.g., newspapers and news platforms. An important aspect of this mission is to provide unbiased information and include diverse perspectives [20]. This helps prevent one-sided or unbalanced news, which can potentially damage the democratic values of modern societies or the reputation of newspapers [5]. Such an editorial mission can ensure diversity and fairness in reporting, which is necessary for maintaining public trust and credibility.

In this paper, we focus on the editorial process carried out daily by editors (and journalists) and propose an *automatic* approach to support them in this process. Our approach is designed as a recommendation tool for editors, assisting them in their daily editorial tasks and routines. Our approach uses state-of-the-art text embeddings to build a representation of the textual content of the news articles, followed by a machine learning classifier that learns from their choices of related articles (as a form of feedback) and incorporates this information for better generating a list of recommendations. While this can be a notable challenge in the news recommendation domain, it has, to our knowledge, received limited attention from the relevant research communities [11].

To achieve that goal, we have formulated the following research questions:

- **RQ1:** Which combination of embedding model and machine learning classifier best predicts the relevance between articles based on editorial feedback?

Authors' addresses: Bilal Mahmood, Bilal.Mahmood@uib.no, MediaFutures, University of Bergen, Bergen, Norway; Mehdi Elahi, Mehdi.Elahi@uib.no, MediaFutures, University of Bergen, Bergen, Norway; Farhad Vadiee, farhad.vadiee@uib.no, MediaFutures, University of Bergen, Bergen, Norway; Samia Touileb, samia.touileb@uib.no, MediaFutures, University of Bergen, Bergen, Norway; Lubos Steskal, Lubos.Steskal@tv2.no, TV 2, Bergen, Norway.

- **RQ2:** What is the best candidate size for generating Top-N recommendations using the best-performing machine learning classifier with various embedding models?

We obtained a comprehensive real-world dataset from TV 2, one of Norway's leading editor-managed commercial media houses. This dataset contains 49,757 news articles, curated with editor-selected related articles. State-of-the-art text embedding models in the Norwegian language were employed to encode the textual content of the news articles and utilized as features to represent the articles.

We evaluated our proposed approach by comparing a set of popular machine learning classifiers against each other to determine the best-performing classifiers in terms of various evaluation metrics. We considered two evaluation scenarios: (i) *classification* and (ii) *recommendation*. In the former scenario, we evaluated the considered classifiers and compared them, while taking into account the embedding models used, in terms of Accuracy, Precision, Recall, and AUC. In the latter scenario, we generated recommendation based on the relevance scores predicted by the classifiers to be suggested to the news editors. We considered Precision@5, Recall@5, and MAP@5 to measure the quality of the recommendations and considered a simpler baseline (similarity-based) following the approach described in [11]. The results of both evaluation scenarios demonstrate the effectiveness of our proposed automatic approach in identifying related articles and generating recommendations for editors to support the editorial process.

The rest of the paper is structured as follows: In Section 3 we review the related work and in Section 4, we describe the methodology used in this work. In Section 5, we discuss the experimental results, and finally, in Section 6, we provide a discussion and conclusion.

3 RELATED WORK

Over the past years, the research on News Recommender Systems (NRSs) has drawn considerable interest from the academic community. This growing attention has explored various approaches employed by online platforms for publishing news, including social networks, and has examined how automated algorithms are extensively utilized alongside editorial moderation.

According to the literature in this domain, significant attention in research has primarily focused on the development of novel algorithms that can effectively analyze different types of user data collected by news platforms, learn the user preferences, and build models to generate recommendations tailored to the specific preferences [12]. Naturally, the main focus of these works has been on improving the metrics of recommendation accuracy from the end-user's perspective. This emphasis has led to the development of a wide range of algorithms aimed at enhancing recommendation quality, primarily based on accuracy-oriented metrics. Most algorithms rely on popular approaches such as content-based filtering and collaborative filtering, each of which is capable of exploiting different types of data, such as content data (e.g., the title and description of the news articles) or user data (e.g., clicks on the news articles) in the recommendation process. Additionally, other algorithms have focused on hybridizing these two approaches to address their respective limitations [8].

While employing novel algorithms is certainly crucial for generating quality recommendations, other factors should also be considered. Such consideration may become particularly important in the news domain, where editorial curation also plays a significant role. Reviewing the literature, a few studies have been conducted to investigate this aspect of the news domain, e.g., by comparing the differences between mechanisms utilized for news selection by editors in comparison to the algorithms, according to the opinions of the audience [18]. A notable example can be the research study that conducted a field experiment to examine the differences in performance between automated recommendations and editor curation [15]. The findings indicated that several factors, including the editors' experience and the quantity of the user data provided to the algorithm, can influence the performance of these approaches.

A limited number of studies have highlighted the multi-stakeholder perspectives, hence highlighting the role of other stakeholders in this domain [1]. News organizations are examples of such stakeholders, which may take additional considerations in the news domain [10], such as editorial values and their responsibility towards the public audience. Other considerations can be public service goals or regulatory requirements [19]. Incorporating all of these considerations into the recommendation process shall result in positive impacts, e.g., the inclusive and fair perspective of the diverse ideas within a democratic society [6]. Indeed, editorially managed news platforms are often regarded as crucial for informing the public about significant societal issues, perhaps a key aspect of democracy, and serving them with responsible news recommendations [3]. The British Broadcasting Corporation (BBC), for example, has set principles for both news publications and TV programs. These principles focus on delivering content that reflects the different cultures of their audience and includes various viewpoints. By adhering to these principles, more comprehensive coverage is maintained, which can resonate with the interests of a wider range of the public [2]. Similarly, policymakers like the Council of Europe have created standards for public broadcasters. These standards require that programs show the cultural and linguistic diversity of their audiences [4]. Such efforts are important for establishing an inclusive media landscape that respects and represents various audiences in a modern society.

It is worth noting that, despite the examples of research mentioned above, it is evident that the editorial aspects of news recommendation have so far received limited attention in the research community. We believe there is a potential need for further research in this field, and this paper aims to address that need. Moreover, the paper differs from these prior research works in various aspects. First, while the majority of existing research primarily focuses on recommending news articles to "users" of the news platforms based on their preferences, we propose a recommendation approach that supports "editors" (and journalists) by providing recommendations to assist them in the selection of related articles. Furthermore, previous studies have largely focused on a *recommendation* scenario only and evaluation based on metrics designed for that specific scenario. In contrast, this paper considers both *classification* and *recommendation* scenarios, utilizing two distinct sets of metrics tailored to each. We believe this dual approach is better aligned with real-world editorial practices, where editors (and journalists) first search for

related articles, narrow down a shortlist of top candidate articles, and then make their final selections from those candidate articles.

4 METHODOLOGY

4.1 Dataset

We received a real-world dataset from TV 2, one of Norway's largest editor-managed commercial media houses. The dataset contains 49,757 news articles published between January 1^{st} 2018 to January 3^{rd} 2023, and for which editors picked at least one related article.

After preprocessing the dataset, which involved dropping invalid articles, we were left with a total of 37,614 valid news articles. On average, each article has a median of two related articles. To prepare the dataset for supervised machine learning, we computed the similarity between news articles and, for each news article, considered the five most similar articles from the past year. This prefiltering step was conducted to avoid the high computational expense of comparing all articles against each other in real-world scenarios, where most articles will expectedly be dissimilar (and unrelated). In addition to that, this step allowed us to focus on the challenging task of identifying related articles within a prefiltered set of similar articles, where not all articles had been selected by editors as related.

Following this step, articles that have not been selected by the editors as related were assigned a target label of 0, while those considered related were assigned a target label of 1. It is worth noting that, our methodology is inspired by the current workflow of the editors in their selection of related articles, where they typically use a search tool to find a short list of candidate articles and then select them as related (or unrelated) based on their domain knowledge and editorial principles. Figure 1 demonstrates this process.

To represent the textual information of the news articles, three different embedding models were considered (and the languages they support): OpenAI's text-embedding-3-small[1] (multilingual), SBERT_{base}[2] (English and Norwegian), and NorBERT3_{large}[3] (Norwegian). Each of these pre-trained models produces different embedding vectors. For instance, OpenAI's embeddings produce a 1,536-dimensional representation, NorBERT3_{large} produces a 1,024-dimensional vector representation of the text, and NB-SBERT_{base} produces 768-dimensional embeddings of the textual information in the news article. Mean pooling of the output layer is used to produce the embeddings for NorBERT3_{large} and NB-SBERT_{base}.

4.2 Evaluation

We have considered a set of popular machine learning classifiers offered by Scikit-learn library [14], i.e., K-Nearest-Neighbors (KNN), Random Forest, and Gradient Boosting. We used standard models, with their default model parameters for training, validating, and testing the performance of different classifiers. As a baseline, we used a Random classifier, that predicted whether the potential news article was related (relevant) or not with an equal probability of 50%.

To train and evaluate the classifiers, we employed a time-based evaluation strategy [7], where all the articles published in 2018 and 2020 were considered for training the classifiers, articles published in 2021 were used for validation purpose, and articles published in 2022 as well as articles in January 2023 were considered for testing. Table 1 presents the dataset characteristics. Since we considered three different embedding models for representing textual information in the news articles, this resulted in three different subsets for training and evaluating different classifiers, as it can be seen in the Table.

In total, there were 34,446 news articles in the training set, 2,932 news articles in the validation set, and 236 news articles in the test set. The prepared training dataset based on OpenAI had 220,248 news article-potential news article pairs, with a feature size of 3,074. Similarly, the training dataset for SBERT and NorBERT3 had 226,225 and 227,266 numbers of news article-potential news article pairs. The different number of training, validation, and test sets across different embedding models is because we used 5 most similar items for target labeling which would result in the different number of 5 most similar items. Figure 2 presents the set of features (denoted as X) as well as the target label (denoted as Y) that were used for training different classification models.

4.3 Metrics

4.3.1 Classification Scenario. To evaluate the effectiveness of our proposed approach, we employed several common evaluation metrics: Accuracy, Precision, Recall, and the Area Under the Receiver Operating Characteristic Curve (AUC). Although we also computed the F1 score, its results were similar to those of the other metrics and are therefore not reported. The formulas for these metrics are presented below.

In the context of this scenario, TP (True Positive) denotes the articles predicted as related by the classifier and selected as related by the editor. FP (False Positive) represents the articles predicted as related by the classifier but not selected as related by the editor. FN (False Negative) refers to the articles predicted as not related by the classifier but selected as related based on the editor's decision. Lastly, TN (True Negative) indicates the articles predicted as not related by the classifier and not selected as related by the editor.

We have considered Accuracy, Precision, and Recall metrics for evaluating the performance of the machine learning classifiers. Accuracy measures the overall correctness of the model, defined as the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision measures the proportion of the true positive predictions among all positive predictions made by the classifier, reflecting its capability to minimize false positives:

$$Precision = \frac{TP}{TP + FP}$$

Recall, also known as Sensitivity, measures the proportion of actual positives correctly identified by the classifier:

$$Recall = \frac{TP}{TP + FN}$$

The Area Under the ROC Curve (AUC) is a widely adopted metric for evaluating the performance of a classifier. It indicates the

[1] https://openai.com/index/new-embedding-models-and-api-updates/
[2] https://huggingface.co/NbAiLab/nb-sbert-base
[3] https://huggingface.co/ltg/norbert3-large

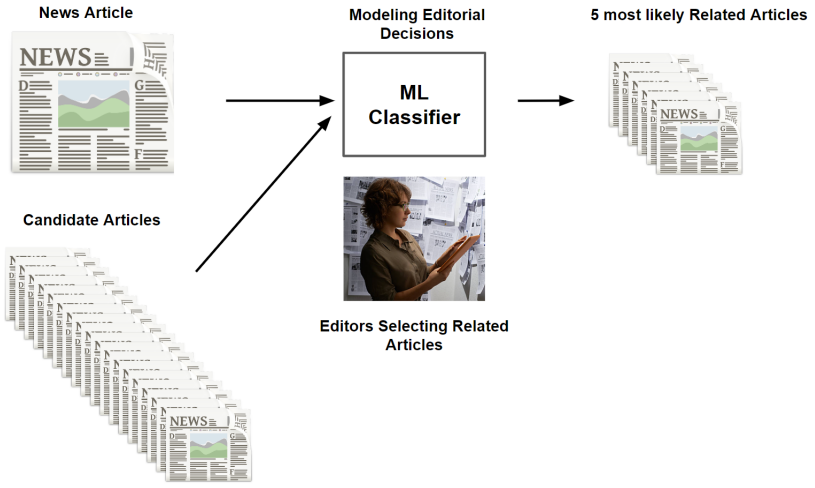


Fig. 1. The schematic view of our proposed (automatic) approach, based on supervised machine learning, capable of supporting editors when selecting related news articles

Table 1. Characteristic of the datasets used for training, validation, and testing the classifiers

Embedding Models	Training Set (2018-2020)			Validation Set (2021)			Test Set (2022)		
	Dimension	#Articles	Related [%]	Dimension	#Articles	Related [%]	Dimension	#Articles	Related [%]
OpenAI	220248, 3074	34446	29.4	17471, 3074	2932	22.9	1372, 3074	236	23.2
SBERT	226225, 1538	34446	28.7	17929, 1538	2932	22.3	1412, 1538	236	22.5
NorBERT3	227266, 2050	34446	28.5	18059, 2050	2932	22.1	1429, 2050	236	22.3

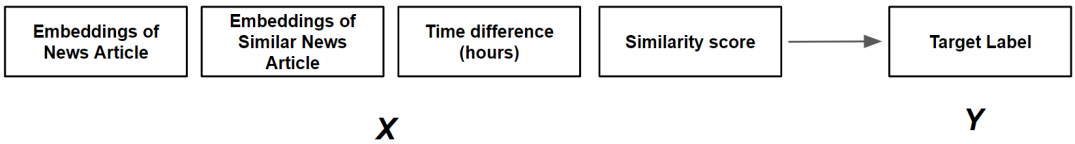


Fig. 2. Features used for training different machine learning classifiers

probability that the classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance. The AUC is calculated based on the following components:

$$Specificity = \frac{TN}{FP + TN}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

Here, Specificity measures the proportion of true negatives correctly identified. Sensitivity (also known as True Positive Rate or Recall) then measures the proportion of true positives correctly identified by the classifier. The AUC combines these two aspects to provide a single scalar value that summarizes the overall performance of the classifier across different threshold values.

4.3.2 Recommendation Scenario. Since we aim to support editors in finding the related articles for a given news article through the recommendation, we considered metrics [16] that can specifically be used to measure the recommendation quality, i.e., Precision@K, Recall@K, and MAP@K, where we considered K = 5.

Precision@K is a common metric that measures the accuracy in recommending relevant items. To compute Precision@K, the top K items are selected for a recommendation for each news article i. Then Precision@K ($P@K$) is calculated as follows:

$$P_i@K = \frac{|L_i \cap \hat{L}_i|}{|\hat{L}_i|}$$

Here, L_i denotes the set of related articles selected by the editor for a given news article i in the test set T, and $\hat{L}_i$ represents the recommendation list containing the top K articles in the candidate set with the highest scores as predicted by the machine learning classifier for the news article i. The overall Precision@K ($P@K$) is then obtained by averaging the $P_i@K$ values across all news articles in the test set.

Recall@K (*R@K*) is another important metric used to evaluate the effectiveness of a recommendation system. For a given news article i, $R_i@K$ is defined as:

$$R_i@K = \frac{|L_i \cap \hat{L}_i|}{|L_i|}$$

In this formula, L_i denotes the set of related articles selected by the editor for a given news article i in the test set T, $\hat{L}_i$ represents the recommendation list containing the top K articles in the candidate set with the highest scores as predicted by the machine learning classifier for news article i. The overall *Recall@K* (*R@K*) is then computed by averaging the $R_i@K$ values across all news articles in the test set.

Mean Average Precision (MAP@K) is a metric that assesses the quality of the ranking in recommendation systems. MAP@K is computed by taking into account the arithmetic mean of the Average Precision@K (AP@K) across all the news articles in the test set. The Average Precision for the top K recommendations ($AP@K$) is calculated as follows:

$$AP@K = \frac{1}{\min(N, K)} \sum_{i=1}^{K} P@i \cdot rel(i)$$

Here, $rel(i)$ is an indicator function that equals 1 if the i^{th} recommended item is related and 0 otherwise. N represents the total number of related articles for a given news article and K is the size of the recommendation list.

5 RESULTS

In addressing our research questions, we conducted a set of experiments focused on the classification of news articles and the prediction of related ones (Experiment A) and then used the predictions to generate recommendations of related news articles (Experiment B). We designed Experiment A to address RQ1, and Experiment B to address RQ2. In this section, we describe the results of these experiments.

5.1 Experiment A: Classification of News Articles Based on Editorial Feedback

We have built and evaluated several well-known machine learning classifiers, i.e., Gradient Boosting, K-Nearest Neighbors (KNN), and Random Forest. Each classifier was trained on the training data, described in the previous Section 4 (Methodology), that have been created based on various embedding models, specifically OpenAI, SBERT, and NorBERT3, to encode the news articles. This approach allowed us to assess the performance of each classifier-embedding combination in accurately predicting the relevance of articles. Our goal was to identify the best combination of classifiers and embedding models for predicting a set of related news articles, based on editorial feedback (i.e., our ground truth). The results of these predictions will subsequently be used for the task of Top-N news recommendations (see Experiment B).

Table 2 presents the results of Experiment A. As can be seen, overall, the best-performing classifier is Gradient Boosting, regardless of the embedding model used, with respect to most of the considered metrics. Additionally, it is noteworthy that the classifiers almost always perform substantially higher than the baseline (random).

Table 2. Comparison of the performances of different classifiers and embedding models on the test set

Embedding Models	Machine Learning	Evaluation Metrics			
		Accuracy	Precision	Recall	AUC
OpenAI	KNN	0.840	0.763	0.447	0.833
	Random Forest	0.802	**0.883**	0.167	0.806
	Gradient Boosting	**0.861**	0.836	0.497	**0.887**
	Random (baseline)	0.511	0.240	**0.513**	0.500
SBERT	KNN	0.865	0.823	0.513	0.850
	Random Forest	0.866	**0.864**	0.481	0.859
	Gradient Boosting	**0.904**	0.861	**0.682**	**0.930**
	Random (baseline)	0.510	0.231	0.506	0.500
NorBERT3	KNN	0.880	0.873	0.541	0.851
	Random Forest	0.876	0.873	0.519	0.894
	Gradient Boosting	**0.910**	**0.886**	**0.686**	**0.936**
	Random (baseline)	0.524	0.243	0.540	0.500

Using the OpenAI model, Gradient Boosting achieved an accuracy of 0.861, while Random Forest and K-Nearest Neighbors (KNN) obtained accuracy values of 0.802 and 0.840, respectively. In terms of precision, however, Random Forest performs the best with a value of 0.883. The precision value is 0.836 for Gradient Boosting and 0.763 for K-Nearest Neighbors. In terms of recall, surprisingly, the Random classifier performs the best with the score of 0.513. Gradient Boosting performs with a score of 0.497. While K-Nearest Neighbors achieves a comparable recall score of 0.447. Strangely, Random Forest does not perform well with respect to this metric, obtaining a value of 0.167. Similarly, in terms of AUC, Gradient Boosting is the best with a value of 0.887, followed by K-Nearest Neighbors with 0.833 and Random Forest with 0.806. For the baseline (Random) classifier, expectedly, the values were lowest for these metrics: 0.511 for accuracy, 0.240 for precision, and 0.500 for AUC.

Considering SBERT as the embedding model, Gradient Boosting yields an accuracy of 0.904, while this value was 0.865 for K-Nearest Neighbors and 0.866 for Random Forest. Comparing the values of precision, Random Forest achieves better results with a score of 0.864, slightly outperforming Gradient Boosting with a value of 0.861. K-Nearest Neighbors achieves a value of 0.823. In terms of recall, the best performing is Gradient Boosting with a value of 0.682. For K-Nearest Neighbors and Random Forest, the recall scores were 0.513 and 0.481, respectively. In terms of AUC, Gradient Boosting is the best with a value of 0.930. The result for K-Nearest Neighbors is 0.850 and for Random Forest is 0.859. For the baseline (Random) classifier, the recorded metrics were as follows: an accuracy of 0.510, a precision of 0.231, a recall of 0.506, and an AUC of 0.500.

When the NorBERT3 embedding model was applied, the results were overall better than the other models for nearly all classifiers. Moreover, for all metrics, Gradient Boosting outperformed the other classifiers. For accuracy, the value for this classifier is 0.910, while for K-Nearest Neighbors it was 0.880, and for Random Forest, it was 0.876. For precision, Gradient Boosting showed a value of 0.886, while both K-Nearest Neighbors and Random Forest showed a value of 0.873. For recall, Gradient Boosting again achieved the best score

with 0.686, while K-Nearest Neighbors and Random Forest obtained 0.541 and 0.519, respectively. Finally, for AUC, Gradient Boosting was the best with 0.936, K-Nearest Neighbors 0.851, and Random Forest 0.894. In the baseline (Random) classifier, again, the values of the metrics were the lowest, with an accuracy value of 0.524, a precision of 0.243, a recall of 0.540, and an AUC of 0.500.

Overall, the results of experiment A present the effectiveness of our approach based on supervised machine learning in predicting whether the potential news articles are related to a given article or not.

5.2 Experiment B: Recommendation of Related News Articles

To address RQ2, we considered a scenario where a new story is published on a news platform, and the editor (and/or a journalist) is going to find a set of related articles for it. These related articles could also be considered editorial suggestions from the perspective of the user who is reading that story on the news platform. Our proposed approach supports the editors by automating this by recommending a set of short-listed candidate articles that an editor can check and select from as related articles. This is achieved by ranking the news articles according to the predicted relatedness scores from the top-performing machine learning classifier (i.e., Gradient Boosting) and then selecting the Top-N articles for recommendation. We consider $N = 5$ meaning that we recommend 5 news articles to the editors. Again, these 5 recommended articles are chosen from a larger set of candidate articles (candidate set), which are the most similar news articles to the target article. The recommendation list is then evaluated against the selections made by editors. We adopted various metrics to evaluate the related article recommendation, i.e., Recall@5, Precision@5, and MAP@5 to measure the quality of the recommendations as described before in the metrics section.

The results of varying candidate sizes are presented in Figure 3. As can be seen, the quality of recommendations can change significantly depending on the size of the candidate set used for generating recommendations with all of the considered embedding models.

When OpenAI and the Gradient Boosting classifier are used (Figure 3-top left), the Recall@5 curve reaches the peak value at the candidate size of 20, and then it starts to decrease steadily. Precision@5 shows similar behavior and peaks at the candidate size of 20. However, MAP@5 reflects a steady decrease with the peak at 5. Looking at NorBERT3 and Gradient Boosting classifier (Figure 3-bottom left) we observe that the peak for the Recall@5 and MAP@5 happens at the candidate size of 5, where the highest value of Precision@5 is reached at the candidate size of 30. Observing the SBERT and the Gradient Boosting classifier (Figure 3-bottom right), we observed that the best value for the Recall@5 and MAP@5 occurs at the candidate size of 5 and Precision@5 at 10.

For the sake of comparison, we also randomly selected articles from the candidate set, which was created using the OpenAI and Gradient Boosting classifiers. The key difference from this baseline is that similarity among the articles was not considered when generating the recommendation list for the editors. The results can be seen in Figure 3-top right, showing a similar trend where the metric values continuously decrease as the candidate size increases, with the best performance observed at a candidate size of 5.

It is important to note that in the recommendation scenario, a crucial consideration was the measurement of the proportion of related articles successfully retrieved within the Top-N recommended articles for the editors. Therefore, we prioritized Recall@5 to choose the best candidate size. The results are presented in Table 3. We also show the baseline results without applying the machine learning classifiers, where the recommendations are generated by only considering the 5 most similar articles based on *Cosine* similarity [9, 13].

Overall, the results indicate that across all three embedding models, the OpenAI embeddings with a candidate size of 20 and the Gradient Boosting classifier yielded the best results in terms of Precision@5 and Recall@5. Interestingly, the MAP@5 results for this model and classifier were comparable to those of the similarity-based method (baseline). Surprisingly, the other embedding models did not show significant differences from the similarity-based method (baseline) in terms of the best candidate size, Precision@5, and Recall@5. However, we observed a considerable improvement in MAP@5.

Table 3. Overall summary of the results for the experiments focused on recommendation scenario with varying candidate sizes.

Embedding Models	Best Classifier	Best Size	Recall@5	Precision@5	MAP@5
OpenAI	Gradient Boosting	**20**	**0.467**	**0.122**	0.361
	Similarity-based	_	0.436	0.107	**0.366**
SBERT	Gradient Boosting	5	0.311	0.073	0.270
	Similarity-based	_	0.311	0.073	0.234
NorBERT3	Gradient Boosting	5	0.238	0.059	0.217
	Similarity-based	_	0.238	0.059	0.187

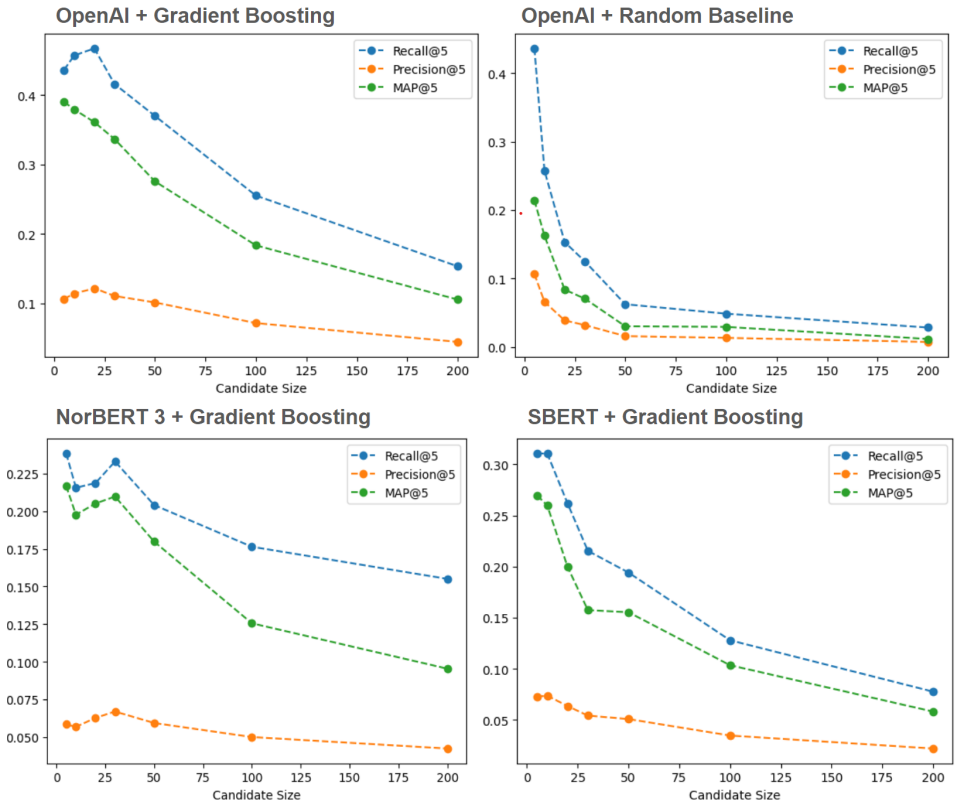
6 DISCUSSION AND CONCLUSION

One of the crucial roles of editors in news platforms is to curate news articles and ensure that related content is appropriately linked to a target news article. This empowers the users of the platforms to better explore a set of *manually* selected articles following the editorial principles, that can also be of users' interest to read further. While this process is important for maintaining the relevance and coherence of news articles on the platform, it can be both expensive and time-consuming, and often requiring significant effort.

In this paper, we propose an *automatic* approach to support editors in this task by utilizing state-of-the-art embedding models to encode the textual content of articles, thereby creating robust vector representations of the news content. These representations are then adopted by a set of popular machine learning classifiers to learn from the data and predict a relevance score, indicating the level of relatedness among articles.

We have evaluated our approach based on a real-world dataset provided by TV 2, one of Norway's largest editor-managed commercial media houses. We considered two evaluation scenarios: (i) a *classification* scenario, where we assessed the accuracy of the classifiers' predictions, and (ii) a *recommendation* scenario, where the output of the classifiers is utilized to generate article recommendations that editors might consider as related content.

Fig. 3. Comparison of the quality of related article recommendations as the size of the candidate set varies for different embedding models and the gradient boosting classifier



We employed several evaluation metrics, including Precision, Recall, and AUC, for the classification scenario, as well as Precision@K, Recall@K, and MAP@K for the recommendation scenario, to comprehensively assess the performance of our approach. The results of our experiments are promising, reflecting the effectiveness of our approach in potentially addressing the essential aspects of the editorial process in the news domain and fulfilling them by accurately classifying articles and recommending related ones.

It is worth noting that, the recommendation scenario we considered in this paper is common in real-world cases which often begins with an editor using some form of search engine to find a set of short-listed candidate articles, among which the most related articles are selected manually. The size of the candidate article set is an important factor in this process since a larger set is expected to increase the chance of finding more related articles. However, this can be computationally expensive as it may require calculating the relatedness against all the articles published in the past. Additionally, conducting such a calculation entirely might be unnecessary since the majority of articles might not be related to each other. Hence, finding a reasonable candidate size can indeed be very beneficial in real-world scenarios. Thus, we report the best candidate size on the test set based on different embedding models and the best performing classifier.

In future work, we plan to add more features about the news articles (e.g., authorship, news categories, entities) for training our classification model. In addition, we plan to further fine-tune the embedding models to improve the quality of the representations generated for the news articles. This could positively impact the accuracy of determining relatedness or similarity between articles. Additionally, we plan to incorporate fine-tuned GPT models and potentially utilize them to rank candidate articles when generating recommendations.

ACKNOWLEDGMENTS

This work was in part supported by the Research Council of Norway with funding to MediaFutures: Research Centre for Responsible Media Technology and Innovation, through the Centre for Research-based Innovation scheme, project number 309339.

REFERENCES

[1] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction*, 30:127–158, 2020.

[2] BBC. Mission, values and public purposes - About the BBC. https://www.bbc.com/aboutthebbc/governance/mission, March 2019.

[3] Elda BROGI, Daniella BORGES, Roberta CARLINI, Iva NENADIC, Konrad BLEYER-SIMON, Jan Erik KERMER, Urbano REVIGLIO DELLA VENARIA, Matteo TREVISAN, and Sofia VERZA. The european media freedom act: media freedom, freedom of expression and pluralism. Technical report, Policy Department for Citizens' Rights and Constitutional Affairs, 2023.

[4] Council of Europe, Commissioner. Public service broadcasting under threat in Europe. https://www.coe.int/en/web/commissioner/-/public-service-broadcasting-under-threat-in-europe, May 2017.

[5] Mehdi Elahi, Dietmar Jannach, Lars Skjærven, Erik Knudsen, Helle Sjøvaag, Kristian Tolonen, Øyvind Holmstad, Igor Pipkin, Eivind Throndsen, Agnes Stenbom, et al. Towards responsible media recommendation. *AI and Ethics*, pages 1–12, 2022.

[6] Natali Helberger. On the democratic role of news recommenders. In *Algorithms, automation, and news*, pages 14–33. Routledge, 2021.

[7] Chip Huyen. *Designing Machine Learning Systems.* O'Reilly Media, USA, 2022.

[8] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. News recommender systems–survey and roads ahead. *Information Processing & Management*, 54(6):1203–1227, 2018.

[9] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, pages 73–105, 2011.

[10] Feng Lu, Anca Dumitrache, and David Graus. Beyond optimizing for clicks: Incorporating editorial values in news recommendation. In *Proceedings of the 28th ACM conference on user modeling, adaptation and personalization*, pages 145–153, 2020.

[11] Bilal Mahmood, Mehdi Elahi, Samia Touileb, Lubos Steskal, and Christoph Trattner. Incorporating editorial feedback in the evaluation of news recommender systems. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pages 148–153, 2024.

[12] Eliza Mitova, Sina Blassnig, Edina Strikovic, Aleksandra Urman, Aniko Hannak, Claes H de Vreese, and Frank Esser. News recommender systems: A programmatic research review. *Annals of the International Communication Association*, 47(1):84–113, 2023.

[13] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*, pages 325–341. Springer, 2007.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] Christian Peukert, Ananya Sen, and Jörg Claussen. The editor and the algorithm: Recommendation technology in online news. *Management science*, 2023.

[16] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, 7:95–116, 2018.

[17] Derek Thompson. How many stories do newspapers publish per day? *The Atlantic*, 2016. Accessed: 2024-08-19.

[18] Neil Thurman, Judith Moeller, Natali Helberger, and Damian Trilling. My friends, editors, algorithms, and i: Examining audience attitudes to news selection. *Digital journalism*, 7(4):447–469, 2019.

[19] Nava Tintarev, Emily Sullivan, Dror Guldin, Sihang Qiu, and Daan Odjik. Same, same, but different: algorithmic diversification of viewpoints in news. In *Adjunct publication of the 26th conference on user modeling, adaptation and personalization*, pages 7–13, 2018.

[20] Christoph Trattner, Dietmar Jannach, Enrico Motta, Irene Costera Meijer, Nicholas Diakopoulos, Mehdi Elahi, Andreas L Opdahl, Bjørnar Tessem, Njål Borch, Morten Fjeld, et al. Responsible media technology and ai: challenges and research directions. *AI and Ethics*, 2(4):585–594, 2022.