Recommendation systems for news articles at the BBC

Maria Panteli
British Broadcasting Corporation
London, United Kingdom
maria.panteli@bbc.co.uk

Alessandro Piscopo
British Broadcasting Corporation
London, United Kingdom
alessandro.piscopo@bbc.co.uk

Adam Harland
British Broadcasting Corporation
Glasgow, United Kingdom
adam.harland@bbc.co.uk

Jonathan Tutcher
British Broadcasting Corporation
Salford, United Kingdom
jon.tutcher@bbc.co.uk

Felix Mercer Moss
British Broadcasting Corporation
Bristol, United Kingdom
felix.mercermoss@bbc.co.uk

ABSTRACT

Personalised user experiences have improved engagement in many industry applications. When it comes to news recommendations, and especially for a public service broadcaster like the BBC, recommendation systems need to be in line with the editorial policy and the business values of the organisation. In this paper we describe how we develop recommendation systems for news articles at the BBC. We present three models and describe how they compare with baseline approaches such as random and popularity. We also discuss the metrics we use, the unique challenges we face and the considerations needed to ensure the recommendations we generate uphold the trust and quality standards of the BBC.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → *Machine learning approaches*.

KEYWORDS

recommendations, news, neural networks

1 INTRODUCTION

The BBC is one of the world's leading public service broadcasters. Its services—television, radio, digital—reach more than 80% of UK's adult population every week [2] and 279 million people worldwide (World Service [4]). This large audience has access to a vast and diverse amount of content, including video, audio and text, spanning topics such as news, sport, and entertainment. In order to enable its audience to enjoy the best possible experience, it is crucial for the BBC to adopt strategies to guide users to the most relevant and engaging content. The main approach until recently has been to manually curate content following the guidelines formally documented in an editorial tome [3]. These have been developed to ensure quality across all products, uphold the BBC values, and build audience trust. Although manual curation is an excellent way to surface quality content, it is not tailored to the user and is hard to scale—the more the amount of content, the harder it is for curators to find relevant items for each type of content. In order to deliver an experience which is relevant, timely, and contextually useful to every single user, the BBC combines editorial curation with personalised, automated approaches. Data-driven recommendations are a key part of these approaches: they are an important tool to enhance users' ability to explore and discover content they would not be aware of otherwise (see e.g. [26, 31, 36, 37]) and have been successfully tested and deployed by several media providers (e.g. Netflix [20]) and e-commerce companies (e.g. Amazon [41]).

According to the mission of the BBC, the organisation must "act in the public interest, serving all audiences through the provision of impartial, high-quality and distinctive output and services which inform, educate, and entertain" [5]. Following this mission, the BBC must be a provider of accurate and unbiased information and the content it produces and distributes must aim to engage diverse audiences. Amongst the diverse types of content produced by the BBC, news is the product that likely contributes most to its reputation as a trustworthy and authoritative media outlet. Besides the UK service BBC News[1], the BBC produces, broadcasts, and delivers online news in more than 40 languages. Hence, it is of utmost importance for automated recommendation approaches implemented on any BBC news service to be not only as accurate as possible, but also to conform with the principles outlined above. This paper reports early results of the experiments we carried out to that end. In particular, it describes the development of recommendation systems for BBC news articles and the challenges in building data-driven applications for a public service broadcaster. The case study adopted in the experiment was the application of recommendation systems for BBC Mundo[2], a Spanish-language news website and part of BBC World Service [6].

The structure of this paper is as follows. Section 2 defines the problem addressed in the current work, and Section 3 discusses prior related work. Section 4 describes the methodology including the data, models, and evaluation approaches. Finally, results are presented and discussed in Sections 5 and 6.

[1] Please note that 'News' capitalised refers to the UK channel, whereas lowercase regards to the type of content.
[2] https://www.bbc.com/mundo

2 PROBLEM DEFINITION

Our goal is to build recommendation systems for news articles. Recommendations in the news domain have been characterised distinctly in the literature [38] due to the short life-cycle of items and the vast amounts of anonymous users. Considering the reputation of the BBC and the responsibility it has to deliver trustworthy and authoritative news to its audience, we highlight the following challenges in achieving our goal.

Non-signed in users. The majority of users on any BBC news platform are not signed in. This means that we have limited information about the user and the items they have previously interacted with. We typically work with session-based information, i.e. user-item interactions that occurred within 30 minutes from each other. This means that our recommendation models need to achieve high accuracy for cold-start user scenarios or predict the user's taste after as little as one item interaction.

Many cold-start items. The publication cycle on any news platform is rapid and unrelenting. BBC News is no different. Fresh items are regularly uploaded and any recommendation system we implement should be able to serve an item within minutes of publication. Additionally, articles may become obsolete or gain sudden relevance following an event—consider for example the case of breaking news. Recommendation approaches must thus be able to take these characteristics into account, not being based solely on a user's history, but considering the content and context of the articles they read.

Architecture constraints. Because of the popularity of BBC news, multiple stakeholders (internal and external) rely on and set the requirements for the news platform. Any changes to the system architecture that could affect other stakeholders need to be thoroughly investigated and justified. Our recommendation models often have to adapt to the existing architecture which means that our system architecture choices are somewhat constrained.

Mistakes are not tolerated. BBC news, and the Mundo platform in particular, are consumed by millions of users. For the majority of these users, this is the only BBC platform they visit. News is also a very sensitive domain as is not just entertainment but is also the way in which people inform and educate themselves. Mistakes in data-driven recommendations could lead to misinformation or compromise our quality standards, something which will largely impact our audience. The bar for the performance of the system is set very high to limit the risk of unexpected behaviour.

Fairness and impartiality. The BBC has built its trust after many years of thoughtful manual curation and expert editorial guidance. It commits to delivering content in a fair, impartial and honest way and data-driven recommendations should live up to, and advance, these standards. Algorithmic fairness and impartiality in recommendation systems are increasingly discussed in the literature [19, 33] but with no standardised solutions yet. We consider evaluation metrics that help us track the risk and bias induced by our recommendation systems.

The above challenges drive the decisions we make around which models and evaluation strategies to implement. For example, we place significant focus upon offline evaluation to avoid unexpected behaviour; we use a variety of metrics to track the quality of recommendations; we consider recency-based systems an essential baseline for news recommendations; and we adopt content-based approaches to tackle the cold-start scenarios. More details about our choices and how they relate to these challenges are provided in Sections 3 and 4.

3 RELATED WORK

Recommendation systems in the news domain have been investigated for more than a decade [27, 38], following various approaches. Collaborative filtering [15] relies on past user behaviour to formulate recommendations based on commonalities across user preferences. Content-based approaches rely on item properties (or user profiles constructed by the properties of the items they consume) to recommend related items [10, 29, 39]. Rather than considering the long user history, session-based approaches focus on user-item interactions that occur within a certain time frame or context [40, 43]. Finally, hybrid systems may put together aspects from these approaches and use a broader range of features, in order to achieve a more nuanced representation of user activity [18, 30]. Content-based, session-based, and hybrid approaches appear to be the most suitable to address some of the problems we outlined earlier, namely the large number of anonymous users and cold-start items (Section 2).

Beyond the news domain, recommendation systems have been investigated in a variety of industrial applications. Approaches vary between traditional content-based and collaborative filtering while, more recently, the advent of deep neural networks has facilitated the development of hybrid strategies [45]. These have been applied to the problem of accommodation search at Airbnb [22], product advertisement at Criteo [28], video recommendations at Youtube [14], and movie recommendations at Netflix [20]. Industry approaches using neural networks are of particular interest to us due to the scalability of the systems and the domain agnostic capability of neural networks.

Considering the system architecture, some neural network-based approaches for recommending textual content are end-to-end (for example [1]), that is, the model takes as input the text of items related to a user, extracts features for the items and the user, and ultimately outputs a recommendation. Other approaches rely on separate modules for extracting features for the content and the user and for generating recommendations [16]. Here, we take the latter approach for a number of reasons. First, an end-to-end approach was not compatible with the current architecture of the system, over which we have limited control (Section 2). Second, separating

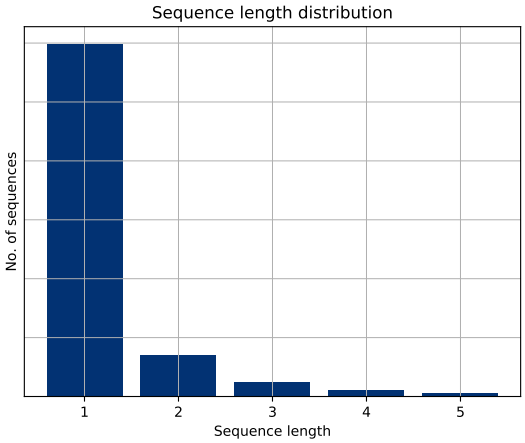

Figure 1: Sequence length distribution in our dataset. The graph includes 99% of sequence lengths, in order to leave out the long tail and improve readability.

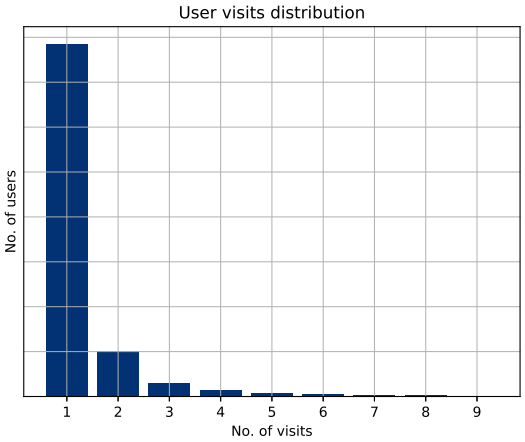

Figure 2: User visits distribution in our dataset. The graph includes 99% of the number of visits, in order to leave out the long tail and improve readability.

content representation from the generation of recommendations enables further experimentation and increases the ability of the system to retrieve new items [16].

4 METHODOLOGY

4.1 Data

The BBC collects detailed user interaction data for its digital services, providing information about users and the circumstances of their visits to BBC websites. For the purpose of this analysis, we used 15-days worth of data from BBC Mundo, spanning from the 6th to 20th April 2019. We define a sequence, or visit, as any succession of user interactions (i.e. page views) within 30 minutes from each other. Page views were aggregated into sequences according to this definition. In this dataset, the average number of user interactions we collected per day was in the order of millions. As shown in Figure 1, most recorded sessions included only a single article read (i.e., a sequence of length 1) which is a common observation in news delivery platforms [16]. Users often visited BBC Mundo only once over the time-span considered (Figure 2).

Like all statistical learning models, to robustly evaluate recommender system performance, the data is required to be appropriately split. In traditional machine learning problems where the raw data takes the form of input-output pairs, this split is relatively straightforward. Assuming there is enough data, a common split might be 80%, 10%, 10% into training, validation and test sets respectively. For recommender systems, the temporal nature of the data makes the situation a little different. While we still need to perform a train/validation/test split, referred to from now on as the test split, we also need to perform an additional split, henceforth be referred to as the query split. The query split describes the process of transforming a temporal sequence of consumption logs into a single or group of feature-target pairs suitable for ingestion into algorithmic learning models.

For the test split, our initial thought was to discard the temporal dimension and sample user sessions according to pre-determined train/test/validation fractions. While the simplicity of this approach is attractive, we decided that to maximise the similarity between our offline testing framework and our online production environment was more important. The temporal approach we implemented is displayed in Figure 3 where we choose a thirteen-day period for training, the next day for validation and the following day for test. As we have the capacity to train and serve fresh consumer-facing models every day, we aim for this offline approach to reflect our production environment sufficiently for inferences in the former to provide valuable information about the latter.

For the query split, we take a user session from a given period defined earlier in the current section and divide it into the maximum number of trigrams while preserving temporal order. Then, for each trigram, the first two elements (articles vectors) represent the user profile while the third and final element is the groundtruth item used as a target for our models. The length of the user profile was chosen based upon two factors: (1) our client-side serving infrastructure is currently limited to providing the current and previous article; and (2) exploratory analysis indicated that minimal gains were made from increasing the number of items that make up the user profile.

4.2 System architecture and models

All recommendation models we implemented were constrained by the need to have compatibility with our current system architecture. This consists of three main components. The first is responsible for generating article embeddings. The second takes user data and article embeddings as input and

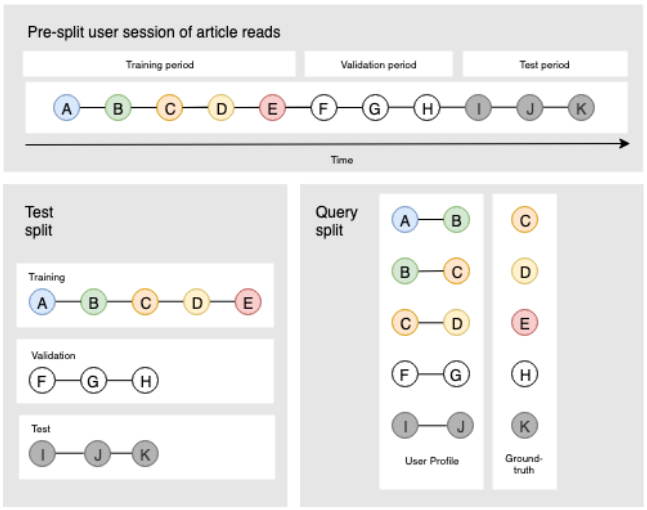

Figure 3: Two splits were performed upon the raw user logs. The test split temporally divided the dataset into train (13 days), validation (1 day) and test (1 day). Then for the query split, each user log session was split into trigrams whereby the first two items represented a user profile and model input while the third represented the groundtruth and model output.

produces a user embedding. Finally, the outputs of the first and the second modules are combined by the third component, which ranks the recommended articles for a user, based on a nearest neighbour search in the latent article space (Figure 4).

The content representation module generates article embeddings. The article embeddings were derived using a Latent Dirichlet Allocation (LDA) model as found performant in related research [9]. LDA is an unsupervised topic modelling approach that represents each document by the probability of a number of topics. The number of topics is defined in advance. Prior work from another BBC team found the optimal number of topics to be 75 for a related dataset of BBC Mundo articles [9].

The user representation module generates user embeddings. The user embeddings are derived from the article embeddings and previous user interactions. Our experiments focused primarily on developing models to derive user embeddings. We explored neural network approaches that combine both content and user data as well as models based only on user interactions (i.e. Cosine-based collaborative filtering model, Section 4.2.2).

The output of the user representation module is subsequently processed by the recommendation generation module. This component takes as input a user embedding and performs an approximate nearest neighbour search in the article latent space, returning as output the K articles with the smallest distance to the user embedding. The distance is computed using the *angular* metric from the Python package ANNOY [7], defined as $\sqrt{2(1 - cos(a, b))}$ for a user embedding a and an article embedding b.

We evaluated three different models to derive the user embeddings: a) a weighted average of item embeddings (Section 4.2.1), b) a cosine-based collaborative filtering method (Section 4.2.2), and c) a rank-optimised neural network (Section 4.2.3). The sections below describe each approach in detail.

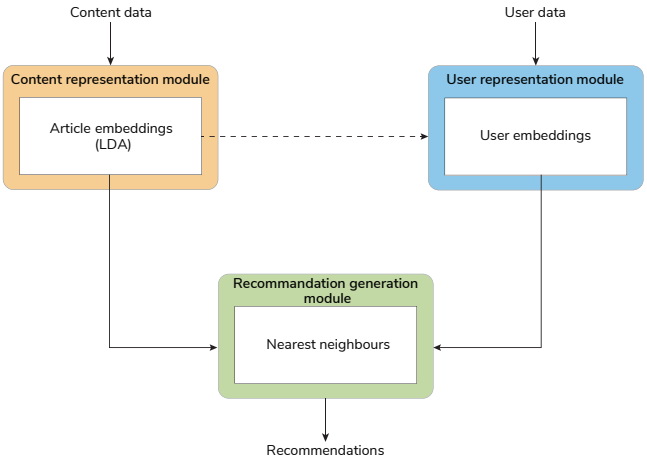

Figure 4: Overview of system architecture and how it relates to the development of user models. A given content representation module provides article embeddings—currently LDA vectors—that are fed into both a user representation module and a nearest neighbour search component. The recommended articles for a user denote the K nearest neighbours to the user vector.

4.2.1 Weighted average of item embeddings. The first user representation model we tested derived the user embeddings from the weighted average of item embeddings, for all items consumed by a given user within a session. The most recently consumed item was weighted by a factor α while the rest of the items in the user's session were weighted by $1 - \alpha$.

4.2.2 Cosine-based collaborative filtering. The second approach was a combination of simple user-item collaborative filtering and a session-based approach. Since users do not need to log in to view the articles, we had no explicit user profile and instead treated each session as a user. To generate the sparse user-item matrix, we took the article IDs for all user sessions within a given time window. The inputs to the model at prediction time were the IDs of the articles viewed in the current user session, and the output was the K highest scored items based on these interactions. Our metric for scoring the articles to recommend was the cosine distance of the current user session and all other user sessions.

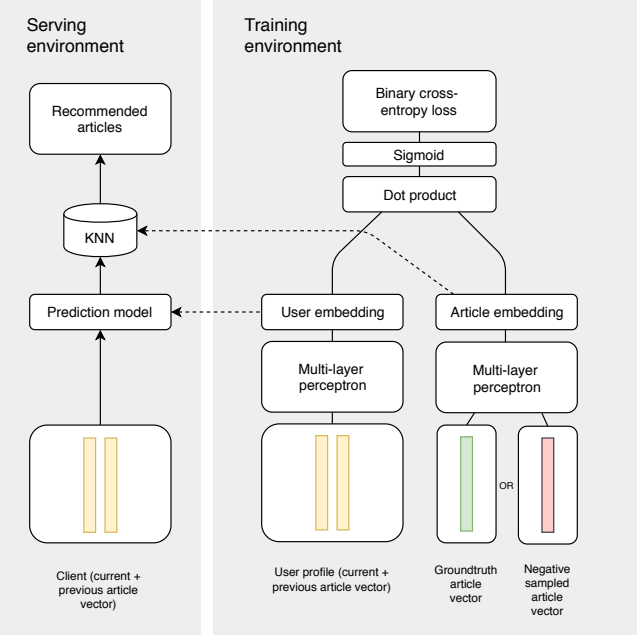

Figure 5: Pointwise neural network architecture for learning-to-rank problem.

4.2.3 Rank-optimised neural network. Motivated by the awareness that a simple linear combination of a user's current and previous article representations led to modest performance gains over using solely the current article, we sought to explore non-linear combinations of these vectors. Artificial neural networks are ideally suited to fitting such non-linear functions, while we were encouraged by the results reported by others that have successfully used deep architectures to solve information retrieval problems, e.g. [11, 14, 22, 46].

The challenge we faced was to design a neural network architecture which learned a latent representation of a user profile (current and previous article) to minimise the distance between itself and the latent representation of the most appropriately recommended article (in this case, the subsequently consumed article). One way of reflecting this problem is a *pointwise* architecture that behaves in a way similar to a regression problem. The model illustrated in Figure 5 takes a user profile (two concatenated 75-length vectors) and an article as input (a 75-length vector), passes each through a five-layer perceptron (with 1024, 512, 256, 128 and 75 hidden units, each with rectified linear activation functions). The model then minimises the binary cross-entropy between the target and the inner product of the final layer of the two perceptrons. Batch normalisation placed before the activation functions of the initial layers was found to significantly boost performance while also halving convergence time, facilitating greater experimentation. Training runs including dropout layers produced no improvement in accuracy so were not included in the final model. Negative articles were randomly over-sampled from the population of positive articles, whereby each training user profile has one positive article and five negative articles. Once this model had been trained, two further models were derived from it for use in the prediction environment. The first, the *user model*, took only the user profile as input and returned the final layer of the connected five-layer perceptron. The second, the *article model*, took only a single article as input and returned the fifth layer of its own five-layer perceptron. The article model was then used to transform all of the raw LDA embeddings into the *article model* embedding space before being fed into our vector-based nearest neighbour index.

4.3 Evaluation

The aim of our work is not only to increase user engagement with BBC products, but also to inform, educate, and entertain—according to the mission of our organisation. We build recommendation systems taking into account these values and develop evaluation strategies that reflect our mission. This section focuses on offline evaluation metrics and the baselines we use in our experiments. Online evaluation is also a big part of our work but goes beyond the scope of this paper which focuses on preliminary results.

4.3.1 Metrics. When developing recommendation models offline, we currently monitor and optimise performance with reference to a suite of six quantitative metrics. For all metrics (with the exception of inter-list diversity) a value can be computed for each groundtruth/recommendations list pair. The overall metric is computed as the mean value over all groundtruth/recommendations list pairs within the test period. For each metric, in addition to calculating the overall value, we also estimate the item-normalised value by first taking the mean metric value for every unique groundtruth item. This value provides an insight into the performance of an algorithm independently of the test set bias towards popular groundtruth items. All metrics were calculated upon recommendation lists of length $K = 100$. We use a relatively large K motivated by the finding that deeper cut-offs in offline experiments provide greater robustness and discriminative power [42] as well as by the fact that we have to exclude a lot of the recommended items a posteriori due to our extensive business rules. A brief description of each metric is provided below (for further details see [12, 21, 34]).

Normalised Discounted Cumulative Gain (NDCG). It measures the gain of a document based on its ranked position in the top 100 list, with lower ranks discounted by a logarithmic factor, and normalises the result by the maximum gain of an ideal top 100 list.

Hitrate. A recall-based metric whereby a recommended list of items is assigned 1 if it contains the groundtruth item, and 0 otherwise.

Intra-list diversity. It estimates the average distance between every pair of items in a recommendations list. For

the experiments reported here, distance between two articles is measured as the ANNOY *angular* distance (described formally in Section 4.2) between two article embeddings.

Inter-list diversity. It measures how diverse the recommended items across multiple lists are. It compares two lists of recommendations and computes the ratio of unique items in these lists over the total number of recommended items between these lists.

Popularity-based surprisal. It measures how novel or surprising the items in a list are. It is formally defined as the log of the inverse popularity of an item (i.e. the probability of observing an item in the recommendations) [12].

Recency. : Measures how recent the recommended items are. It calculates the time difference between the recommendation request and the age of the recommended items using a Gaussian decay function. The mean is set to 1 and the standard deviation is chosen such that articles of 7 days old or more receive a score less than 0.5.

The ideal recommendation engine would optimise all these metrics providing recommendations that are relevant to the user, but that are also diverse, recent, and avoid the popularity bias. In practice this is usually a trade-off as an algorithm that provides more accurate results is, conversely, less likely to produce diverse ones (and vice versa). In line with our values and objectives, we sometimes choose algorithms that favour diverse and recent content at the cost of a certain degree of accuracy.

4.3.2 Baselines. We compare our user models to four baseline approaches and require that each new user model outperforms the existing ones. We consider the following recommenders as baselines:

- *Random recommender*: Produces K random recommendations.
- *Recency-based recommender*: Ranks item by recency and returns the top K most recent items.
- *Popularity-based recommender*: Ranks items by popularity and returns the top K most popular items.
- *Content similarity recommender*: Finds the K nearest neighbours of an item (e.g., the last item consumed by a user) using the ANNOY *angular* distance between item embeddings.

Our offline experiments report results on the four baselines defined in above and the three models defined in Section 4.2. We use the NDCG metric to comment on the accuracy of the systems and the remaining metrics defined in Section 4.3.1 to comment on qualitative aspects of the recommendations.

5 RESULTS

The NDCG scores for each recommender system are shown in Figure 6. The scores from all metrics are summarised in Table 1.

Accuracy scores recorded for the baselines models were in line with expectations. Compared to a random selection of

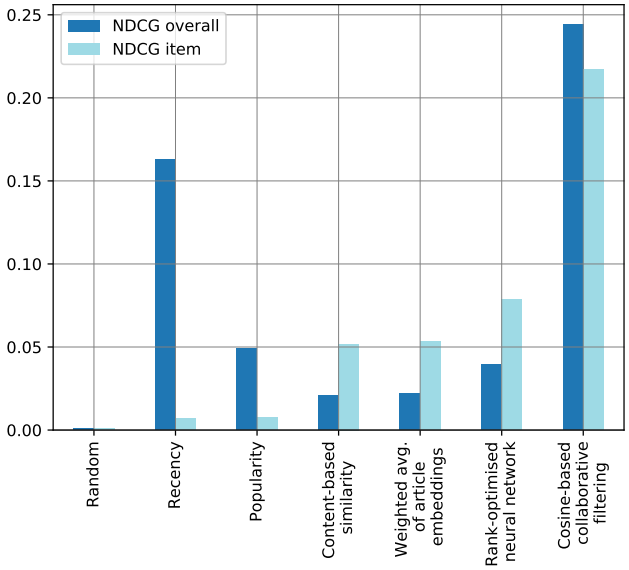

Figure 6: Overall and item-normalised NDCG for the four baselines described in Section 4.3.2 and the three user models described in Section 4.2.

items (the *random* model), all other baselines show clear performance improvements for both overall and item-normalised NDCG. The *popularity* and *recency* recommenders returned higher values than the *content-based similarity* (CS) model for NDCG overall; however, if the most popular items are factored out by looking at the item-normalised score, the opposite is true. The *recency* recommender scored particularly high NDCG overall which confirms our expectation that users in a news platform prefer to consume fresh content.

Of the implemented models, the *cosine-based collaborative filtering* (CF) model (Section 4.2.2) outperformed all baselines and other models by a significant margin, this being the case both for overall and item-normalised NDCG and hitrate. However, this significant advantage in accuracy comes at a cost to inter-list diversity and surprisal, where both other models returned higher scores. However, this effect was not observed with the intra-list diversity metric, indicating that individual CF lists contained more diverse content while the lists of the other models were more distinct.

The *weighted average* (WA) model (described in Section 4.2.1, with α optimised at 0.7) achieved accuracy scores surpassing all the baselines in item-normalised NDCG, although as expected, this was not the case for NDCG overall. This suggests that the model consistently projects into relevant regions of the embedding space, and that the nearest neighbours are not just most popular candidates. Despite returning marginally higher NDCG scores, the WA results are salient mainly for how similar they are across the board, to the CS baseline that lacks information from the previous article.

The *rank-optimised neural network* (NN) model (Section 4.2.3) returned accuracy scores that were a clear step up from both

Table 1: Benchmark results of competing models after generating 100-length lists of recommendations. For the sake of brevity, we report here only overall metrics.

Recommender System	Hitrate	NDCG	Intra-list diversity	Inter-list diversity	Surprisal	Recency
Random baseline	0.005	0.001	**1.192**	**0.995**	0.430	0.010
Recency baseline	0.695	0.163	1.175	0.000	0.000	**0.975**
Popularity baseline	0.315	0.049	1.170	0.000	0.000	0.495
Content similarity baseline	0.085	0.021	0.641	0.968	0.790	0.018
Weighted average of item embeddings	0.065	0.022	0.641	0.968	**0.790**	0.018
Cosine-based collaborative filtering	**0.741**	**0.244**	1.154	0.584	0.480	0.512
Rank-optimised neural network	0.128	0.040	0.909	0.731	0.781	0.036

other LDA-based models (CF and WA). This was the case for both variants of NDCG and particularly so for hitrate, indicating that the NN model was optimised more for recall than precision and could possibly benefit from further reranking procedures. The NN model also distinguished itself from CF and WA models in the diversity and surprisal metrics. Results suggest the NN model produces more distinct lists (indicated by higher inter-list diversity) but that those lists are more topically homogenous (indicated by lower intra-list diversity and surprisal metrics).

6 DISCUSSION

The first cycle of research in our journey to find the best news recommender for BBC Mundo is complete. In Section 2 we have outlined the characteristics of the problem we address: a majority of non-signed in users; a large number of cold-start items; architectural constraints; and high quality demands, not only in terms of accuracy, but also in what concerns fairness and impartiality of recommendations.

One of the lessons we learned is that—unsurprisingly—balancing the different aspects of our problem is hard. One model may satisfy one of our requirements, whilst failing to fulfil another. A pure collaborative filtering approach is currently our best option to maximise offline scoring accuracy, but that comes at the cost of reducing diversity (and a degree of recency, dependent upon how regularly we retrain). Moreover, the performance of the CF model was not entirely unexpected, as it has been shown [17, 25, 32] that such simple methods typically outperform the neural approaches when only logged user items are used, and instead only start to perform well when the input features contain additional contextual meta-data. However, as with most collaborative filtering approaches, this model suffers from the item cold-start problem and so frequent generation of the user-item sparse matrix would be required. Therefore, we cannot depend upon a solution that is derived purely from user interactions. To that end, we also know from our experiments that the contribution of previous articles appears to have a lower impact than expected. Despite performance of the WA model consistently exceeding the CS baseline model (across validation and test), this gain was always marginal. Furthermore, our attempts at combining the current and previous article vectors in a non-linear fashion using a neural network had an impact that was also weaker than expected. These unintuitive results raise further questions that we plan to explore in the future.

Fundamentally, we believe there is scope to optimise the NN approach further so that it will perform more competitively with CF. To achieve this end we have multiple strategies. These fall into three categories: model architecture, data, and training improvements.

We know that learning to rank in a pointwise framework is not optimal. Both pairwise and listwise approaches should, in theory, achieve better results (see [13, 23]). Pairwise loss functions together with triplet loss architectures have demonstrated impressive results elsewhere but our own early experiments have indicated they are difficult to train, tending towards significant underfitting.

A key reason for this may be the under-representation of negative examples in our training set. Adopting a higher proportion of negative training examples may address this, but also using more informed negative sampling techniques may be required (such as weighted approximate-rank pairwise loss [44]). Even with the current pointwise architecture there is a 5% difference in train/test performance (item-normalised NDCG) that should significantly reduce by using the appropriate regularisation.

Changes to our training process may also lead to significant gains. In addition to increasing compute resources for the exploration of the hyperparameter space, reducing the training/testing window from the order of days/weeks to the order of hours may provide greater scope for experimentation (as has been reported elsewhere [16]). While a smaller training window does necessitate more regular training of deployed models, it also means more manageable datasets where hyperparameter optimisation is more practical.

A further change that may prove fruitful is to expand the richness of the input to the user profile model. This may include expanding the size of the user journeys in the training set beyond 3 (a constraint which, incidentally, did not apply to the CF model at training), while also introducing contextual information about the user.

Finally, another direction to be explored in the future regards content representation. In experiments not reported in the current work, raw article text has been encoded through an LDA model. However, our system architecture affords

enough flexibility to replace the current content model with alternative article embeddings and test different approaches. In particular, we are interested in taking sub-word information into consideration [8], enriching text with semantics [10, 24], and augmenting text representations with multimedia [35, 46].

Our results demonstrate the difficulty of acquiring all the desired characteristics of an ideal news recommender. Ultimately, we expect ensemble approaches may represent the best solution. Here we may take the cold-start benefits of the content-based neural approach and combine it with the less diverse but more accurate list of items generated by a collaborative filtering model.

7 CONCLUSION

In this paper we evaluated three approaches to provide news recommendations for the BBC Mundo service. The systems we have built are compatible with BBC serving infrastructure, a use case which includes millions of daily users and new content in the order of several thousand articles per week. In spite of our experiment being only the initial step of a journey that promises to be much longer, our models outperformed random, popularity-based, recency-based and content-similarity baselines. It is worth noticing though, that these results do not reflect current online performance. More work is needed to ensure these models, when deployed, meet the quality and editorial standards of the BBC. Future challenges do not concern only achieving higher accuracy, but also conforming to the principles of algorithmic fairness and impartiality. We encourage the community to collaborate in helping us create the way forward towards fair and engaging recommendations and applications with responsible machine learning.

REFERENCES

[1] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. *Ask the GRU*: Multi-task Learning for Deep Text Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*. 107–114.

[2] BBC. 2019. The BBC's services in the UK - About the BBC. https://www.bbc.com/aboutthebbc/whatwedo/publicservices Consulted on 21 June 2019.

[3] BBC. 2019. Editorial Guidelines. https://www.bbc.co.uk/editorialguidelines Consulted on 21 June 2019.

[4] BBC. 2019. Global news services - About the BBC. https://www.bbc.com/aboutthebbc/whatwedo/worldservice Consulted on 21 June 2019.

[5] BBC. 2019. Mission, values and public purposes - About the BBC. https://www.bbc.com/aboutthebbc/governance/mission Consulted on 21 June 2019.

[6] BBC. 2019. News - Mundo. https://www.bbc.com/mundo Consulted on 21 June 2019.

[7] E Bernhardsson. 2017. ANNOY: Approximate nearest neighbors in C++/Python optimized for memory usage and loading/saving to disk. *GitHub https://github. com/spotify/annoy* (2017).

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *CoRR* abs/1607.04606 (2016). arXiv:1607.04606 http://arxiv.org/abs/1607.04606

[9] Clara Higuera Cabañes, Michel Schammel, Shirley Ka Kei Yu, and Ben Fields. 2019. Human-centric Evaluation of Similarity Spaces of News Articles. In *42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (NewsIR'19 Third International Workshop on Recent Trends in News Information Retrieval)*. 51–56.

[10] Michel Capelle, Flavius Frasincar, Marnix Moerland, and Frederik Hogenboom. 2012. Semantics-based news recommendation. In *2nd International Conference on Web Intelligence, Mining and Semantics, WIMS '12, Craiova, Romania, June 6-8, 2012*. 27:1–27:9.

[11] Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. 2018. Word2Vec Applied to Recommendation: Hyperparameters Matter. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 352–356. https://doi.org/10.1145/3240323.3240377

[12] P Castells, S Vargas, and J Wang. 2011. Novelty and diversity metrics for recommender systems: choice, discovery and relevance. In *International Workshop on Diversity in Document Retrieval (DDR 2011) at the 33rd European Conference on Information Retrieval (ECIR 2011)*.

[13] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On Sampling Strategies for Neural Network-based Collaborative Filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. 767–776.

[14] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, 191–198. https://doi.org/10.1145/2959100.2959190

[15] Abhinandan Das, Mayur Datar, Ashutosh Garg, and Shyamsundar Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*. 271–280.

[16] Gabriel de Souza Pereira Moreira. 2018. CHAMELEON: a deep learning meta-architecture for news recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*. 578–583.

[17] Gabriel de Souza Pereira Moreira, Dietmar Jannach, and Adilson Marques da Cunha. 2019. Contextual Hybrid Session-based News Recommendation with Recurrent Neural Networks. *CoRR* abs/1904.10367 (2019).

[18] Elena Viorica Epure, Benjamin Kille, Jon Espen Ingvaldsen, Rébecca Deneckère, Camille Salinesi, and Sahin Albayrak. 2017. Recommending Personalized News in Short User Sessions. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*. 121–129.

[19] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. *CoRR* abs/1905.01989 (2019).

[20] Carlos A. Gomez-Uribe and Neil Hunt. 2016. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans. Management Inf. Syst.* 6, 4 (2016), 13:1–13:19.

[21] Asela Gunawardana and Guy Shani. 2009. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research* 10 (2009), 2935–2962.

[22] Malay Haldar, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C. Turnbull, Brendan M. Collins, and Thomas Legrand. 2018. Applying Deep Learning To Airbnb Search. *CoRR* abs/1810.09591 (2018).

[23] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[24] Wouter IJntema, Frank Goossen, Flavius Frasincar, and Frederik Hogenboom. 2010. Ontology-based news recommendation. In *EDBT/ICDT Workshops (ACM International Conference Proceeding Series)*. ACM.

[25] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*. 306–310.

[26] Tomonari Kamba, Krishna Bharat, and Michael C. Albers. 1996. The Krakatoa Chronicle: An Interactive Personalized Newspaper on the Web. *World Wide Web Journal* 1, 1 (1996).

[27] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. 2018. News recommender systems - Survey and roads ahead. *Inf. Process. Manage.* 54, 6 (2018), 1203–1227.

[28] Romain Lerallut, Diane Gasselin, and Nicolas Le Roux. 2015. Large-Scale Real-Time Product Recommendation at Criteo. In *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*. 232.

[29] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*. 125–134.

[30] Lei Li, Li Zheng, Fan Yang, and Tao Li. 2014. Modeling and broadening temporal user interest in personalized news recommendation. *Expert Syst. Appl.* 41, 7 (2014), 3168–3177.

[31] Greg Linden. 2011. Eli Pariser is wrong. http://glinden.blogspot.com/2011/05/eli-pariser-is-wrong.html Consulted on 21 June 2019.

[32] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Model. User-Adapt. Interact.* 28, 4-5 (2018), 331–390.

[33] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a Fair Marketplace: Counterfactual Evaluation of the trade-off between Relevance, Fairness & Satisfaction in Recommendation Systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. 2243–2251.

[34] Tomoko Murakami, Koichiro Mori, and Ryohei Orihara. 2007. Metrics for Evaluating the Serendipity of Recommendation Lists. In *JSAI (Lecture Notes in Computer Science)*, Vol. 4914. Springer, 40–46.

[35] Thomas Nedelec, Elena Smirnova, and Flavian Vasile. 2017. Specializing Joint Representations for the task of Product Recommendation. *CoRR* abs/1706.07625 (2017). arXiv:1706.07625 http://arxiv.org/abs/1706.07625

[36] Nicholas Negroponte. 1996. *Being Digital*. Random House Inc., New York, NY, USA.

[37] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren G. Terveen, and Joseph A. Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*. 677–686.

[38] Özlem Özgöbek, Jon Atle Gulla, and Riza Cenk Erdur. 2014. A Survey on Challenges and Methods in News Recommendation. In *WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies, Volume 2, Barcelona, Spain, 3-5 April, 2014*. 278–285.

[39] Michael J. Pazzani and Daniel Billsus. 2007. Content-Based Recommendation Systems. In *The Adaptive Web (Lecture Notes in Computer Science)*, Vol. 4321. Springer, 325–341.

[40] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Comput. Surv.* 51, 4 (2018), 66:1–66:36.

[41] Brent Smith and Greg Linden. 2017. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing* 21, 3 (2017), 12–18.

[42] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2018. On the robustness and discriminative power of information retrieval metrics for top-N recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*. 260–268.

[43] Shoujin Wang, Longbing Cao, and Yan Wang. 2019. A Survey on Session-based Recommender Systems. *CoRR* abs/1902.04864 (2019).

[44] Jason Weston, Hector Yee, and Ron J. Weiss. 2013. Learning to rank recommendations with the k-order statistic loss. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*. 245–248.

[45] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1 (2019), 5:1–5:38.

[46] Lu Zheng, Zhao Tan, Kun Han, and Ren Mao. 2018. Collaborative Multi-modal deep learning for the personalized product retrieval in Facebook Marketplace. *CoRR* abs/1805.12312 (2018). arXiv:1805.12312 http://arxiv.org/abs/1805.12312