Semantic and Lexical Token Based Vectors Improve Precision of Recommendations for TV Programmes

Taner Cagali*, Hadi Wazni†, Saba Nazir†, Mehrnoosh Sadrzadeh*
Department of Computer Science,
University College London, U.K
Email: taner.cagali.20@ucl.ac.uk, hadi.wazni20@ucl.ac.uk,
saba.nazir.19@ucl.ac.uk, m.sadrzadeh@ucl.ac.uk

Chris Newell ‡
British Broadcasting Corporation
United Kingdom
Email: chris.newell@bbc.co.uk

***Abstract*—Advances in the digitalisation of data have led to large archives of content in media companies. These archives include multimodal data and metadata associated with each media programme. Relating content across different mediums of data and metadata has thus become an emergent challenge, with applications to popular domains such as programme recommendation. In this paper, we worked with combinations of content similarity measures computed from the distances between different forms of textual data obtained from subtitle files and metadata obtained from the genres of programmes. The different forms of textual representations we considered were neural semantic and topic vectors, and a weighted Jaccard distance encoding lexical token rareness. The late fusion combination of these four distances provided the best recommendation results. For a weekly dataset of 145 TV programmes, it increased the precision of the genre-based recommendations by 5.76%. In a monthly dataset of 906 programmes, it achieved an increase of 1.5%. This combination was more efficient than one with audio and video files.**

***Index Terms*—Neural Embeddings; Semantic Vectors; Topic Models; Jaccard Distance; Rareness; Content; Genre; Cosine Similarity; Hybrid Recommender Models**

I. Introduction

Since digitisation of data in the late 20th century, media programme archives have reached their largest volumes, leading to challenges in content discovery and reuse. Many programmes have content that is related to others, e.g. documentaries are related to news article and drama programmes created on similar themes. A user that watches one programme, might be interested in other related programmes. Modelling and reasoning about content across its different media of data and metadata has thus become a priority in research in recommender systems and the aim of this paper is to move it forward. Our specific research question was whether the distances between content vector representations are good measures for relating content and thus producing precise recommendations. More formally, if programmes p_1 and p_2 each have a set of vector representations $\overrightarrow{p}_1^i$ and $\overrightarrow{p}_2^j$, for $i, j \in \{\text{data1,data2,metatadata1,metadata2}\}$, that are individually close to each other, i.e. $\overrightarrow{p}_1^1$ is close to $\overrightarrow{p}_2^1$ and similarly for other values of i and j, then how likely is it that a viewer who watched p_1 will also watch p_2? In order to find an answer, our first aim was to generate recommendations using the average of the distance measures between $\overrightarrow{p}_1^i$ and $\overrightarrow{p}_2^j$, then calculate the precision of the results against audience behaviour. Our second aim was to improve on the vector representations in order to increase this precision. For recommendation generation, we used a k-Nearest Neighbour algorithm. For improving the quality of the vectors, we followed two strategies: firstly, we used neural semantic and topic vectors with cosine similarity. Secondly, we modelled a lexical token overlap and computed a weighted Jaccard distance. We tested the predictions of the models, on two weekly and monthly TV programme datasets provided to us by the BBC[1].

BBC's archive of content includes data and metadata about each programme. The data can be text, e.g. subtitle files, or audio/video files. The metadata can be the hierarchal genre of the programme, e.g. ⟨drama, soap⟩, and ⟨entertainment, comedy⟩, as well as the cast and the service and channel information. In previous work [1], [2], presented at former IEEE ISM's, we focused on multimodal forms of content and worked with vector representations of text, as well as audio and video files, and genre, service, cast and channel as metadata, on a small weekly dataset of TV programmes. Processing audio and video files became time and resource inefficient as we moved to the large monthly dataset examined in this paper. We discovered that the increase in precision of the recommendations produced by audio and video was similar to the increase when both token and semantic vectors were considered for text.

II. Content Representations

A. Doc2vec

Word2vec [3] was proposed to learn distributed representations of words in vectors, by exploiting the distributional hypothesis [4], which states words that occur in the same context tend to convey similar meanings. Word2vec has two architectures [5]. The first is called Continuous Bag-of-Words (CBOW), it learns word embeddings by trying to predict the centre or target word over a fixed context window. To this end, CBOW maps each word to a unique vector. The concatenation or average of the vectors are used as features to predict the target word. The second architecture, called skip-gram, follows the fundamental structure of CBOW. It instead attempts to

[1]BBC is in the process of anonymising and making some of this data available to public by request.

predict the centre word given the context words. In practice, both architectures incorporate just a single hidden layer.

For many tasks it would be useful to have distributed representations for sentences, paragraphs, and even whole documents. Le and Mikolov expand on Word2vec with paragraph vector [6], or more commonly known as Doc2vec. As the case of Word2vec, Doc2vec consists of two architectures, conceptually different, but computationally very similar. They are Paragraph Vector – Distributed Bag-of-Words (PV-DBOW) and Paragraph Vector – Distributed Memory (PV-DM). Very much like CBOW, PV-DM attempts to predict the centre word based on the context words. Although, it also includes a paragraph embedding (or document embedding) within the prediction process. Therefore, PV-DM is able to simultaneously learn document and word embeddings. PV-DBOW is a much simpler model, as it ignores the context words and attempts to predict a set of randomly sampled words from document. As a consequence, it is capable of learning word and document embeddings.

B. Neural Topic Model

Since the introduction of the Variational Autoencoder (VAE) [7], there has been significant advancement within the field of topic modelling. A notable example is Neural Topic Model (NTM) introduced by Ding et al. [8]. The encoder $q_\phi(z|x)$ is an inference model that serves to compress the bag-of-words representation $x \in \mathbf{R}^{|V|\times 1}$ to a latent space $z \in \mathbf{R}^{K\times 1}$, where $|V|$ is the vocabulary size and K is the number of topics. The decoder is a generative model that represents the likelihood $p_\theta(x|z)$, which attempts to reconstruct z sampled from the encoder. The variational parameters ϕ serves as the weights and biases of the encoder, whereas model parameters θ are the weights and biases of the decoder. The objective function of NTM is to maximise the evidence lower bound (ELBO).

Ding et al. build upon NTM by incorporating a topic coherence aware training objective. Topic coherence measures the interpretability of a topic by estimating the degree of semantic similarity between the top-N words within the topic. Topic coherence or interpretability can be measured by normalized point-wise mutual information (NPMI). Ding et al. approach of constructing a topic coherence training objective leverages pre-trained word embeddings, as they carry contextual similarity information that is highly related to the mutual information terms involved in the calculation of NPMI [8]. The topic coherence regularisation is defined as:

$$T = E^T W \qquad S = ET \qquad C = \sum_i (S \odot W)_i$$

where $E \in \mathbf{R}^{|V|\times D}$ is the pre-trained word embedding matrix for the vocabulary, $T \in \mathbf{R}^{D\times K}$ is the W-weighted centroid topic vector, and $S \in \mathbf{R}^{|V|\times K}$ is the cosine similarity matrix between the word and topic vectors. The objective function of NTM now becomes:

$$\mathcal{L}_R(x;\theta,\phi) = \mathcal{L}_{\text{ELBO}} + \lambda \textstyle\sum_i C_i \tag{1}$$

where λ a hyper-parameter thats controls the strength of topic coherence regularization. Ding et al. name this model NTM-R.

C. Jaccard and Weighted Jaccard

The Jaccard coefficient was introduced to measure the degree to which two sets of tokens agree with each other. It is also known as the *intersection over union* coefficient, as for two sets of tokens A and B it is computed as follows:

$$\frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Jaccard coefficient, also known as Jaccard distance, has been widely used in different fields, including Information Retrieval to compute a degree of token overlap in documents. A document can be modelled as the set of the lemmatised versions of its words, taken to be tokens. The degree of overlap between two documents, is obtained by counting how many tokens they share versus how many they don't. This is easily calculated by computing the Jaccard distance between their set of tokens.

The original form of Jaccard, introduced above, is a good measure for original binary vectors or binarised versions of sets of tokens. If a token repeats multiple times in a set or different tokens have different weights, with some tokens repeating more than others or some being *more valuable* than others, Jaccard does not perform as well. It treats all tokens on equal grounds and ignores the frequencies and weights. Different weighted versions of Jaccard have thus been introduced to take these factors into account and offer an improved distance measure.

We are interested in measuring to what degree the "rare" words of two documents overlap, where rareness is computed over the corpus (set) of documents. For a token w in a corpus C, its degree of rareness Δ_w is defined by the inverse of the frequency of its occurrence in the corpus $\text{freq}(w, C)$, normalised by the total number of tokens in the corpus $\text{total}(C)$, see below. If the weight associated with $a \in A$ is Δ_a and $b \in B$ is Δ_b, their weighted Jaccard coefficient are as follows:

$$\Delta_w = \left(\frac{\text{freq}(w, C)}{\text{total}(C))}\right)^{-1} \qquad \frac{\sum_{c\in A\cap B} \Delta_c}{\sum_{c\in A\cup B} \Delta_c}$$

The number of times a rare word is repeated in two documents is also important when measuring their degree of lexical similarity. Weighted Jaccard can be modified to take this second set of weights into account in a variety of ways, such as taking the *minimum* or *addition* of the frequencies. The formulae for these are given below:

$$\textbf{Min Weights} \qquad \frac{\sum_{c\in A\cap B} \min\{\text{freq}(c, A), \text{freq}(c, B)\} \times \Delta_c}{\sum_{c\in A\cup B} (\text{freq}(c, A) + \text{freq}(c, B)) \times \Delta_c}$$

$$\textbf{Add Weights} \qquad \frac{\sum_{c\in A\cap B} (\text{freq}(c, A) + \text{freq}(c, B)) \times \Delta_c}{\sum_{c\in A\cup B} (\text{freq}(c, A) + \text{freq}(c, B)) \times \Delta_c}$$

The above operations had a similar performance, but **Min Weights** did slightly better. We worked with a few other ways of taking frequencies into account, e.g. multiplication and maximum, but they did not work well. We looked at an indirect way of computing token overlap were cosine similarities between rare words were considered. This also did not work well.

TABLE I: Hybrid Token+Semantic results on the Weekly Dataset

Model	MAP	NDCG	ILD	Surprisal	Personalisation	Coverage
Genre	12.77%	25.23%	52.72%	0.30	76.59%	100%
User	18.51%	34.29%	80.90%	0.19	51.65%	87.94%
Jaccard	11.55%	23.78%	78.77%	0.30	76.46%	100%
PV-DM	13.88%	27.26%	80.37%	0.30	77.43%	100%
NTM-R	15.68%	29.48%	77.74%	0.29	71.15%	100%
Genre+PV-DM+Jaccard	16.74%	31.08%	61.97%	0.30	77.50%	100%
Genre+NTM-R+Jaccard	17.46%	31.91%	63.03%	0.29	74.75%	100%
Genre+PV-DM+NTM-R	18.48%	33.38%	68.42%	0.29	73.85%	100%
Genre+PV-DM+NTM-R+Jaccard	18.57%	33.43%	66.32%	0.29	74.58%	100%

III. Recommender Framework

Recommendations are generated using the k-Nearest Neighbours (k-NN) algorithm. In this approach the prediction value of an item is produced based on two stages. Within the first stage of the algorithm, i.e. the similarity computation stage, the similarity between all item pairs is computed. In this case, the similarity between two items i and j is measured by the cosine of the angle between them. Next, the prediction computation stage, a relevancy score indicating the user's interest is estimated for all items. Recommendations can then be made based on the rank of the relevancy scores. The prediction of item i for the target user u is given by equation (2). In effort to combine the various content representations, a weighted scheme is adopted. The prediction is computed as a linear combination of the relevancy scores and set weights. Suppose that there are m recommendation approaches, the prediction $P_{i,u}$ can be determined as in equation (3):

$$P_{i,u} = \sum_{j \in N_u^K(i)} \text{sim}(i,j) \quad (2), \qquad P_{i,u} = \sum_m w_m P_{i,u}^{(m)} \quad (3)$$

where $N_u^K(i)$ is the k most similar items to i user u has watched and w_m denotes the weight of recommender $P_{i,u}^{(m)}$.

IV. Experimental Results

A. Data and Baselines

We compiled two datasets from the BBC iPlayer, a video on demand service from the BBC, server logs to evaluate the proposed model. The first was composed over a two week period, and the second over two months; the details of these datasets are presented in Table II.

TABLE II: BBC iPlayer Training and Testing Sets

Dataset	Split	#Programmes	#Users	#Viewings
Weekly	Training	145	233,958	1,390,540
	Testing	141	10,000	47,707
Monthly	Training	906	1,068,531	17,984,925
	Testing	631	10,000	59,582

A viewing for a programme is determined based on implicit feedback. Specifically, we identify a programme to be of positive preference if a user views it for longer than five minutes. The reasoning for this is because the during the first few minutes the number of viewers decreases rapidly, but then stabilises at around five. Additionally, the users within the testing set are a subset of the users within the training set.

To gauge our results we also implement two baseline recommenders: (1) content metadata, whereby each programme is expressed as a binary feature vector representing genres, and (2) user viewings from the training sets, also leading to binary item-user feature vectors.

B. Results

In order to appraise the quality of the content representations, the similarity degrees are evaluated both individually and fused, at a rank of 20. Tables I and III present the results for the weekly and monthly datasets. We compare the performance of the recommenders in terms of accuracy: mean average precision (MAP) Normalised Discounted Cumulative Gain (NDCG) [9], and diversity: Intra-list diversity (ILD), surprisal, personalisation and coverage [10], [11], [12].

In previous work [1], [2], we explored the effect of multimodal programme representations. For text, we worked with semantic Doc2Vec and neural topic model vectors. For audio, we worked with a bag of audio word model trained on the audio features provided by the library LibROSA. For video we employed a Res-Net 152 model pre-trained with ImageNet [13], a Res-Net 50 model pre-trained with Places365 [14], and a VGG19 model pre-trained with FER-2013 [15]. The increases we obtained on a late fusion of these vectors for the weekly dataset are presented in Table IV. In the rubric G stands for Genre, T for Text, A stands for Audio, and V for Video.

TABLE IV: Multimodal Evaluations for Weekly Dataset

Model	MAP	NDCG	ILD	Surprisal	Personalisation	Coverage
V+A	11.45%	23.83%	76.91%	0.28	69.01%	100.00%
T+A	17.24%	31.75%	77.26%	0.29	71.81%	100.00%
T+V	17.32%	31.79%	77.05%	0.29	71.62%	100.00%
T+A+V	17.41%	31.95%	76.83%	0.29	71.52%	100.00%
T+A+V+G	18.35%	33.28%	72.47%	0.29	72.71%	100.00%

In previous IEEE ISM's, we also worked with audio and video files of programs [1], [2]. Processing these, however, took orders of magnitude longer than text files and needed access to a different set of computing resources such as GPU processors. Further, storing audio and video assets of programmes took a large amount of memory and came with its own copyright

TABLE III: Hybrid Token+Semantic results on the Monthly Dataset

Model	MAP	NDCG	ILD	Surprisal	Personalisation	Coverage
Genre	2.64%	7.42%	23.17%	0.35	90.96%	96.99%
User	4.64%	11.84%	79.15%	0.18	72.18%	73.53%
Jaccard	2.74%	7.46%	71.93%	0.27	74.99%	88.11%
PV-DM	3.29%	8.48%	59.58%	0.34	93.71%	99.84%
NTM-R	3.28%	8.63%	59.69%	0.35	94.17%	100%
Genre+PV-DM+Jaccard	4.05%	10.03%	49.20%	0.32	91.79%	99.37%
Genre+NTM-R+Jaccard	4.10%	10.17%	48.87%	0.33	91.71%	99.68%
Genre+PV-DM+NTM-R	3.96%	9.89%	48.14%	0.34	93.17%	100%
Genre+PV-DM+NTM-R+Jaccard	4.14%	10.20%	50.60%	0.33	92.44%	99.84%

issues. As a result, we could only take them into account for the weekly dataset (and not the monthly dataset). On the other hand, the precision of the hybrid audio/video model T+A+V+G is 18.35%, which is slightly below the precision of the hybrid token-based and semantics model at 18.57%. We conclude that in the absence of enough resources, different representations of text is a good replacement for different forms of media.

An analysis of specific examples revealed interesting results. A programme such as `Top Gear` which has both an <entertainment> and <factual, car & motor> genre, got recommendations across all genres. Amongst the top programmes returned by Doc2Vec was `Man on the Moon`, which is <drama, bibliographical>, for NTM-R we had a more interesting range. Most of them had a factual genre, e.g. `Raindeer Family and Me`, whose genre is <factual, science&nature>. Jaccard, however, also recommended `Snail and Whale`, which is <children, drama> and `Ski Sunday`, which is <sport>.

V. Conclusion and Discussions

This paper focused on enhancing content discovery within media program archives, to help users discover new content or content not easily visible or accessible. We worked with the subtitle files encoded in semantic and token-based lexical vectors. For semantics, we worked with Doc2Vec and a neural topic model. For tokens, we worked with the degree of rareness and lexical frequencies of the overlapped tokens. The average of distances between these vectors increased the precision of genre-based recommendations. Hybridising all the feature vectors, including a binary genre vector, provided us with a significant increase in precision for two TV datasets.

For rareness, we worked with a weighted Jaccard measure. Rareness is a special case of an information theoretic degree of significance, known as TF-IDF, which was more time consuming and did not perform as well; at rank 20, its MAP was 8.02%, significantly lower than all of our models. A limitation of the current approach is our use of a KNN recommender system, which we are planning to replace with a deep neural network classifier. The KNN system was used by the BBC R&D and that is why used it too.

The combination of token-based and semantic feature vectors slightly surpassed the precision of recommendations when audio and video files were used. Given the extra resources processing audio and video requires and recent advances in Natural Language Processing, we find this result a promising direction to pursue. We aim to improve the quality of our textual embeddings using state-of-the-art transformer-based language models such as BERT and GPT.

References

[1] S. Nazir, T. Cagali, M. Sadrzadeh, and C. Newell, "Audiovisual, genre, neural and topical textual embeddings for tv programme content representation," in *2020 IEEE International Symposium on Multimedia (ISM)*, 2020, pp. 197–200.

[2] T. Cagali, M. Sadrzadeh, and C. Newell, "Enhancing personalised recommendations with the use of multimodal information," in *2021 IEEE International Symposium on Multimedia (ISM)*, 2021, pp. 186–190.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[4] Z. S. Harris, "Distributional structure," 1981.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[6] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *ArXiv*, vol. abs/1405.4053, 2014.

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2014.

[8] R. Ding, R. Nallapati, and B. Xiang, "Coherence-aware neural topic modeling," in *EMNLP*, 2018.

[9] C. Manning, P. Raghavan, and H. Schütze, "Xml retrieval," in *Introduction to Information Retrieval*. Cambridze University Press, 2008.

[10] B. Smyth and P. McClave, "Similarity vs. diversity," in *International conference on case-based reasoning*. Springer, 2001, pp. 347–361.

[11] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, 2010.

[12] M. Ge, C. Delgado-Battenfeld, and D. Jannach, "Beyond accuracy: evaluating recommender systems by coverage and serendipity," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 257–260.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[14] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.

[15] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee *et al.*, "Challenges in representation learning: A report on three machine learning contests," in *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part III 20*. Springer, 2013, pp. 117–124.