Description Based Visual Recommender System for Podcasts

Lukáš Marek
Ústav matematiky, informatiky a kybernetiky
Vysoká škola chemicko-technologická v Praze
Prague, Czech Republic
mareku@vscht.cz

***Abstract*—**
Because of the increasing popularity of podcasts, with new podcasts appearing everyday, there is a greater interest in recommender systems for podcasts. Those allow users to look for new podcasts to listen to and help creators reach larger audiences. We use the Latent Semantic Analysis to find similarity between podcasts using the short description provided by the podcast's author(s). Output of this method is vector representation of each podcast, which is then projected onto 2D plane to create a visual recommender system. ***Index Terms*—NLP, Recommender Systems, Neural Networks, LSA**

I. Introduction

The popularity of podcasts is on steep rise with more listeners than ever before [1, 2]. There is mostly likely no human activity or hobby for which there is no related podcast. Thus it is of interest to be able to search for podcasts based on their content and themes. This is where recommender system come into play. Recommender systems are a information processing systems aimed at recommending products or services to users. The main driving force for development and research, both theoretical and practical, of recommender systems was the rapid expansion of the internet [3]. This trend was further increased by today's focus on collection and analysis of large amount of data (so called Big Data [4]) generated by the users of the internet. Current recommender systems use combinations of filtering and model building to give complex recommendations based on the user's past activity. In contrast to these complex systems, we focus in this work on the building of a simple visual recommender system for podcasts using their description.

II. Dataset

Most of our dataset was downloaded from Podcasts Index website [5]. The data was then prepossessed and only information relevant to this project was extracted (e.g. title, description and iTunes categories). In total, out dataset consisted of $21{,}167$ podcasts. This dataset was then filtered out for non-english podcasts and for missing values, thus bringing the total count down to $14{,}211$. Further preprocessing was done for each method specifically.

III. Methods

A. Latent Semantic Analysis

We approach the problem by analyzing the text descriptions included with each podcast using the methods of *Latent Semantic Analysis* (LSA) [6, 7]. LSA uses a dimensionality reduction of term-occurrence matrix to better capture semantic structure of documents. The dimensionality reduction is usually achieved through Singular Value Decomposition (SVD), rendering the LSA implementation very easy. Because LSA uses *bag of words* approach for the text analysis, a lot of semantic information (word order and context) is lost during analysis. Nevertheless, LSA is still relevant to this day ([8]) and we use it here to assess the dataset quality and as a benchmark for other methods.

To effectively use the aforementioned method, we exclude all podcasts with description shorter than 200 characters and duplicates (resulting in 9400 podcasts). Tokens (words) in each description are then lemmatized (reduced to their canonical form), thus creating a vocabulary containing $32{,}342$ tokens. This vocabulary is then used to create a term-occurrence matrix where each token is given a weight based on its term frequency–inverse document frequency (tf–idf). Tf-idf works on the premise, that terms appearing in fewer texts are assumed more likely to contribute to their meaning (thus are given higher weight) while the contribution of common terms (terms like: *the, be, to* or *and*) is negligible. This makes tf-idf a powerful tool for text-based recommender systems [9].

On the resulting term-occurrence matrix A of size 9400×32342 we perform SVD dimensionality reduction to reduce its size to 100 rows. Each column $\mathbf{a}_i$ thus represents a podcast as a 100 dimensional vector and we use these vector representations to define a similarity between podcasts (using some metric such as cosine similarity or correlation). As a last part of the analysis we embed the vectors into 2D space using *t-distributed stochastic neighbor embedding*.

B. Siamese Neural Network

Deep neural networks (DNN) constitute one of the main pillars of current machine learning. Their ability to work as universal approximators allowed them to be applied in every domain of human endeavour including similarity analysis. More specifically, similarity between two inputs $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{X}$

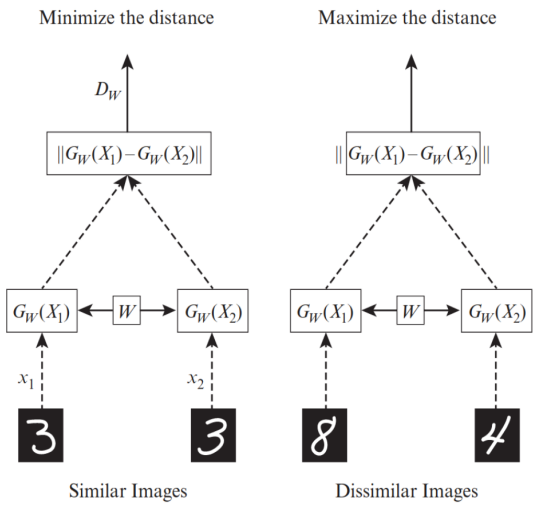

Fig. 1. Example of Siamese Neural Network. Inputs X_1, X_2 are fed into the same network G_W. The difference between the outputs is measured using some distance metric $\|\cdot\|$. The goal then is to maximize the distance between dissimilar inputs and minimize it for similar inputs. Figure taken from [10].

can be described by a similarity metric $d(\mathbf{x}_i, \mathbf{x}_j)$. Our goal then is to learn this metric d. Very often the metric is not applied directly to the inputs, but to their embedding $f(\mathbf{x}_i), f(\mathbf{x}_j)$, where $f : \mathbb{X} \to \mathbb{R}^n$ is embedding function. When f is deep neural network we are then talking about *deep metric learning* [10]. Siamese neural networks [11] are a type of DNN used for *deep metric learning*, which processes multiple inputs (usually two or three) using the same set of neurons. The results for each input are compared and the neural network weights are then adjusted to reflect the similarity between the inputs (**fig.**1). In natural language processing, Siamese neural networks were successfully applied to learning word/term meanings [12] and sentence similarity [13, 14]. In our work we use the model described in [13] as a starting point.

Since we are trying to process a text input (i.e. a sequence of tokens) it is natural to use Recurrent Neural Network as our model. More specifically, our model begins by embedding the podcast description into a sequence of vectors using the *word2vec* word embedding algorithm. [15, 16]. After the embeddings, next part of our model is the LSTM layer ([17, 18]) which processes the sequence one by one until arriving at a final output vector representing the podcast description. The same process is done for some other podcasts and its description to arrive at a second vector representation. These vector representations are then compared by some metric $g : \mathbb{R}^n \to \mathbb{R}$ to get a 'distance' between them. This distance is then compared with the metric we are trying to learn and the weights are then adjusted according some loss function we are trying to minimize.

IV. Results

The performance of out methods was evaluated by two criteria. First was its recommendation ability: Is the method able to find similarities between podcasts? Does the method give good recommendations? etc. The second metric of performance was the visual part of the system (i.e. 2D projection of the vector representations). Here we asked: Are similar podcasts close to each other? Does the 2D projection correctly reflect the recommendations? Are there some larger themes appearing (e.g. podcasts about wrestling and football are close to each other and thus creating a large theme of sport related podcasts)? These criteria are mostly qualitative and thus the result are highly subjective.

A. Latent Semantic Analysis

To evaluate the recommendation ability we perform a k-nearest neighbor (k-NN) search using the correlation metric. Correlation is defined as

$$\delta(\mathbf{u}, \mathbf{v}) = 1 - \frac{(\bar{\mathbf{u}} - \mathbf{u}) \cdot (\bar{\mathbf{v}} - \mathbf{v})}{\|\bar{\mathbf{u}} - \mathbf{u}\|_2 \|\bar{\mathbf{v}} - \mathbf{v}\|_2}$$

where $\bar{\mathbf{u}}$ is the average over the values in $\mathbf{u}$. In the k-NN search, correlation metric and cosine similarity metric both gave almost identical result, but correlation metric gave better result when used for the 2D projection and thus was preferred. In the table I we see two examples of resulting recommendations. First prompt was the *Huberman lab* podcast. Here we can clearly see a sensible answer where almost all recommended podcasts have either health, performance or biology as a theme. On the other hand, the example of the *Smartless* podcast shows poor quality of the recommendations where most recommended podcasts are music related despite the fact that the *Smartless* podcast is not about music. Since both examples only use the podcast description, the recommendations are also vastly different from other platforms (e.g. Spotify, iTunes). On these platforms recommendations are most likely done using the popularity of podcasts and how likely will listeners of one podcast also listen to some other podcast. Because our dataset is absent of this information we left to only use descriptions to access similarity. Overall prompting for podcasts with clearly defined theme, the recommender system was able to find other podcasts with the same or similar theme. However, for podcasts where no overarching theme can be found the recommendations were either completely random or all from unrelated theme (e.g. the description of *Smartless* podcast reads: "*smartless with jason bateman, sean hayes, & will arnett is a podcast that connects and unites people from all walks of life to learn about shared experiences through thoughtful dialogue and organic hilarity. a nice surprise: in each episode of smartless, one of the hosts reveals his mystery guest to the other two. what ensues is a genuinely improvised and authentic conversation filled with laughter and newfound knowledge to feed the smartless mind.*" Here we can clearly see that the description does not contain any major theme and almost all recommended podcast were about music; see **Tab.** I).

TABLE I
EXAMPLE OF THE 9-NEAREST NEIGHBOR SEARCH THROUGH THE LSA VECTOR REPRESENTATIONS OF PODCASTS.

Prompt:	huberman lab	smartless
1.	global health	the vanished podcast
2.	eat, live & move with miyagi	psychedelic psoul
3.	inner cosmos with david eagleman	all songs considered
4.	the persuasion lab with martin medeiros	found
5.	workflow with steve glaveski	death by music podcast
6.	optimal neuro—spine	big band bash
7.	genetics	chat d'or music club
8.	treating trauma podcast	hiddentracks
9.	chemistry for the future: strange substances and structures	hall of songs

For the 2D projection we tried *Principal Component Analysis* but the result was one large cluster of point and hence the T-distributed stochastic neighbor embedding was used. In the figure 2 the 2D projection is plotted (interactive plot can be constructed using the supplementary code at GitHub). We can clearly see a small clusters of commonly themed podcasts on the outside (e.g. we can easily find a cluster of Disney themed podcasts or wrestling podcasts). Unfortunately, the clusters are distributed randomly and there is mostly no association between them (which is to be expected since LSA is unable to 'pickup' semantic similarity between words). One exception is a collection of clusters on the left (yellow rectangle in fig. 2) with Business, Investing and Money themes. Main drawback of this projection is large 'blob' of podcasts in the middle, where no smaller clusters can be found which clearly indicates the inability of LSA to find any common themes. There is definitely a variety of factors at play including the limitations of LSA, poor description of podcasts and in part also themeless podcasts (e.g. *Smartless* above).
In the end, the performance of LSA was better than expected for such a simple method. Its ability to recommend podcasts with very specific main theme is sufficient and improvements in this area using the description only would be mostly subjective. Nevertheless, there is still a significant portion of podcasts where its performance is suboptimal and it is this area where we tried using the Siamese neural networks to get better performance.

B. Siamese Neural Network

Our goal with the Siamese neural network is to increase the performance of the LSA in cases where the text description is not sufficient. We do this by including the information about the iTunes categories for each podcasts which is included in our dataset. First, we introduce some notation. Let p be a podcast from our dataset and denote by $\text{cat}(p)$ the iTunes categories for p. For two podcasts p, q we then define our metric as

$$\mu(p,q) = \delta(p,q) \cdot \frac{|\text{cat}(p) \cap \text{cat}(q)|}{|\text{cat}(p) \cup \text{cat}(q)|}.$$

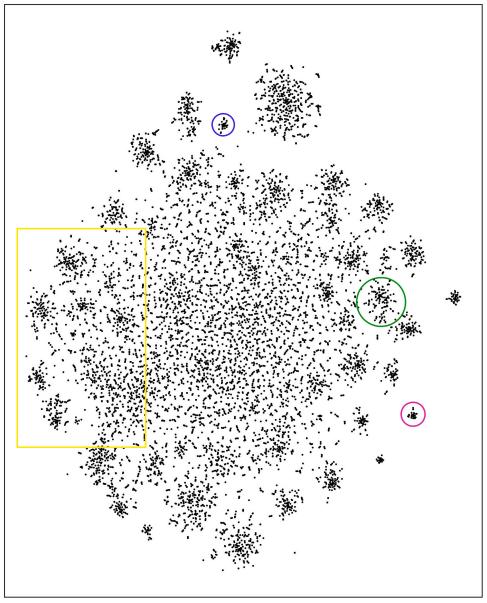

Fig. 2. 2D projection of the latent semantic vector representations of podcasts. Several clusters are identified: wrestling (blue), Disney (magenta) and sports (green). There is also example of commonly themed clusters indicated by the yellow rectangle.

The goal of this metric is to use the LSA representations of podcasts, compute their correlation and adjust it using the shared number of categories.
We begin with the transformation of the input text represented as sequence of tokens into a sequence of vectors $\mathbf{x}_{i=1}^{n} \subseteq \mathbb{R}^d$, where d is the embedding dimension ($d = 64$ in our case). For this we use the skip-gram model from the *word2vec* word embedding algorithm using the podcasts descriptions as a corpus. Alternative to this approach would be using a pretrained model, trained on some much larger corpus of text, with subsequent retraining to include the out-of-vocabulary words. We chose the first approach mainly for educational reasons.
The resulting variable length sequences were fed into a LSTM layer to produce a final 32-dimensional vector representation $\mathbf{h}$ of the given text input. Since we are building a Siamese network, the input always consisted of a pair of podcasts descriptions $\mathbf{x}_1, \mathbf{x}_2$, whose resulting vector representations $\mathbf{h}_1, \mathbf{h}_2$ were compared using the function:

$$g(\mathbf{h}_1, \mathbf{h}_2) = \exp(-\|\mathbf{h}_1 - \mathbf{h}_2\|_1).$$

Finally, the output $g(\mathbf{h}_1, \mathbf{h}_2)$ was compared with $\mu(p,q)$ using the Mean squared error loss function. The trained LSTM layer was then used to get a vector representation for each podcast. Compared to LSA, the recommendation ability of the Siamese network is mostly random (**tab.** II), even for podcasts where LSA performed well. As for the 2D projection (**fig.** 3) the resulting projection gives much larger number of clusters with more sharply defined edges. This is unfortunately a sign of poor performance, because it is not to be expected

that podcasts can be clustered by theme in such a sharply defined groups. After further examination, it was confirmed that no meaningful learning was done by the network, since the clusters mostly consisted of random, unassociated (by humanly distinguishable theme) podcasts. The poor performance this is partly because of the choice of the metric μ and possibly due the simplicity of the model. For future work, the best course of action will be to replace the metric μ by the correlation metric, and then adjust the model to the point where it is able to replicate the results of the LSA. After that, the metric μ can be slowly perturbed to include the iTunes categories into its evaluation. Since this approach is very time and resource consuming, we did not explore it much further.

TABLE II
EXAMPLE OF THE 9-NEAREST NEIGHBOR SEARCH THROUGH THE VECTOR REPRESENTATIONS OF PODCASTS GIVEN BY THE SIAMESE NN.

Prompt:	huberman lab	smartless
1.	the must read alaska show	marketing today with alan hart
2.	shiny happy people with vinay kumar	christian natural health
3.	living wholehearted podcast with jeff and terra	yes catholic
4.	the sacred donut	back 2 life
5.	not nutrition gurus	infinite lunchbox
6.	the power of young people to change the world	down to sleep (audiobooks & bedtime stories)
7.	courage to be seen with sherrie clark	how to do drugs
8.	the therapy show with lisa mustard	unscripted with nell daly
9.	chemistry for the future: strange substances and structures	master self love

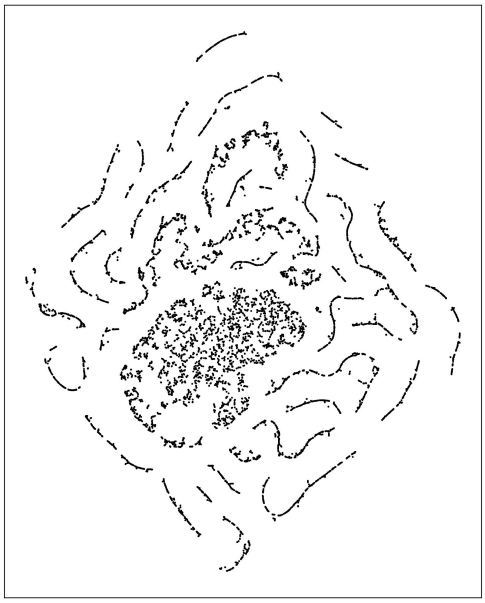

Fig. 3. 2D projection of the Siamese NN representations of podcasts. The only positive feature is the aesthetics. Otherwise there was no learning done.

V. CONCLUSION

In conclusion, we were able to create a very basic recommender system for podcasts, with semi-working visual 2D representation. This was done using very simple method (LSA) and thus demonstrating (again) its power in classification of text documents. Our attempts to improve on the results of this method, using a Siamese neural network, were mostly unsuccessfully and there still remains a great need for further exploration and investigation.

REFERENCES

(1) Gray, C. Podcast Statistics & Industry Trends 2024: Listens, Gear, & More, en, 2024.

(2) 2021 Podcast Stats & Facts (New Research From Apr 2021), en-US, 2017.

(3) Lü, L.; Medo, M.; Yeung, C. H.; Zhang, Y.-C.; Zhang, Z.-K.; Zhou, T. *Physics Reports* **2012**, *519*, 1–49.

(4) Sagiroglu, S.; Sinanc, D. In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp 42–47.

(5) Podcastindex.org, en, 2024.

(6) Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; Harshman, R. *Journal of the American Society for Information Science* **1990**, *41*, 391–407.

(7) Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; Deerwester, S.; Harshman, R. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '88*, ACM Press: Washington, D.C., United States, 1988, pp 281–285.

(8) Evangelopoulos, N.; Zhang, X.; Prybutok, V. R. *European Journal of Information Systems* **2012**, *21*, 70–86.

(9) Beel, J.; Gipp, B.; Langer, S.; Breitinger, C. *International Journal on Digital Libraries* **2016**, *17*, 305–338.

(10) Kevin P. Murphy, *Probabilistic Machine Learning: An introduction*; MIT Press: 2022.

(11) Bromley, J.; Bentz, J. W.; Bottou, L.; Guyon, I.; Lecun, Y.; Moore, C.; Säckinger, E.; Shah, R. *International Journal of Pattern Recognition and Artificial Intelligence* **1993**, *07*, 669–688.

(12) Neculoiu, P.; Versteegh, M.; Rotaru, M. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, Association for Computational Linguistics: Berlin, Germany, 2016, pp 148–157.

(13) Mueller, J.; Thyagarajan, A. *Proceedings of the AAAI Conference on Artificial Intelligence* **2016**, *30*, DOI: 10.1609/aaai.v30i1.10350.

(14) Hu, X.; Zheng, Z.; Li, H.; Wang, W. *2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)* **2022**, 1270–1273.

(15) Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality, en, 2013.

(16) Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space, en, 2013.
(17) Hochreiter, S.; Schmidhuber, J. *Neural Computation* **1997**, *9*, 1735–1780.
(18) Understanding LSTM Networks – colah's blog.