Content-based Recommendation for Podcast Audio-items using Natural Language Processing Techniques

Zhou Xing, Marzieh Parandehgheibi, Fei Xiao, Nilesh Kulkarni and Chris Pouliot
NEXTEV USA, INC.
3200 N 1st St
San Jose, CA 95134
joe.xing, marzieh.parandeh, fei.xiao, nilesh.kulkarni, chris.pouliot@nextev.com

ABSTRACT

A podcast combines the liveliness of a FM radio channel with the economy of internet blog posting. They are especially convenient for scenarios when there is limited internet ability and connectivity for example in the car, the gym, etc. While both the volume and heterogeneity of content is huge it becomes operationally difficult to manually categorize or tag these audio items, thus manage them in a system for users to discover. Furthermore, due to the incompleteness of audio associated meta data there are not enough features for a typical recommender system to learn the item similarities thus make recommendations. In this paper we propose and examine a novel approach to generate latent embeddings for podcast items utilizing the aggregated information from all the text-based features associated with the audio-items. These embeddings that are generated using well established Natural Language Processing (NLP) techniques for the podcast items can be used to measure or indicate the content similarity among the various podcast items. Both GPU (CUDA) and CPU computing architectures are experimented and bench marked for the model training, cross-validation of the content predictions on large scale datasets.

1. INTRODUCTION

A podcast combines the liveliness of a FM radio channel with the economy of internet blog posting. They are especially convenient for scenarios when there is limited internet ability and connectivity. For example, a huge amount of podcasts are consumed when we are staying in the car, the gym, etc. While both the volume and heterogeneity of content is huge with sources of music, audio books, comedy shows, news, etc., it becomes operationally difficult to manually categorize or tag these audio items, thus manage them in a system for users to discover. Furthermore, due to the incompleteness of audio associated meta data such as ID3 tags (genres, artist, etc.) there are not enough features for a typical recommendation system to learn either item similarities (content based) or latent embeddings of like-minded users (collaborative filtering), thus assisting them in exploring the items that are potentially interesting. In this paper we propose and examine a novel approach to generate latent embeddings for podcast items utilizing the aggregated information from all the text-based features associated with the audio-items. These embeddings for the podcast items are then be fed, as input features, to either machine learning or statistical models, for example, a recommendation system. Another unique contribution of this paper is that we have heavily used Natural Language Processing (NLP) techniques to learn fundamental word embedding for Chinese characters

that consist of the podcast descriptive text corpus. This opens up an interesting connection between the nontrivial task of Chinese NLP and recommendation system. There are some well-known shortcoming and un-robustness in Chinese NLP such as the same word may have both different syntactic and semantic meaning, for example, the difference between ordinal numbers, "三年", and numbers with measure nouns, "三年" [1]. Studies on using word embedding to build similarity metrics between podcast objects could shed some light on interesting NLP research. A few techniques that are well exploited in the field of NLP are used to build the fundamental word embeddings such as Conditional Random Field (CRF) based tokenizer for Chinese word segmentation, Skip-gram Negative Sampling (SGNS) method for learning the context of individual Chinese word based upon the given training text corpora from podcast source. Once all these context and subtle relationship between the words are learned, we can aggregate them, as indicators on the type, category, genre, etc. of the podcast audio items. And with other available features such as popularity, temporal length we can build a content or similarity based recommender for podcast items.

2. MODEL FOR WORD EMBEDDINGS

In field of NLP words or phrases from a vocabulary are usually mapped into a vector format of real numbers in a relatively low dimensional manifold (few hundreds dimensions) as compared to the vocabulary size. These vectors are fundamental building blocks for advanced tasks in language processing and understanding that involves scrutinizing the context and meaning of words. Moreover since these word representations do not treat individual words just as unique symbols or atomic symbols (one-hot encoder) but instead reflect similarities and dissimilarities between them, it allows for a higher level aggregation from words to any abstracted granularity such as phrases, documents, even any objects that are associated a particular word set. This embedding process also removes data sparsity and reveals more relationship between different text segments.

[1]the same word can both mean "the third year" or "three years" in this case

2.1 CRF based tokenizer

Since Chinese words are not separated by spaces in a sentence, we use a Chinese word segmenter that was built using a conditional random field (CRF) sequence model [1]. CRF is a discriminative undirected probabilistic graphical model and in this statistical sequence model, the probability assigned for a particular label sequence is

$$P_\lambda(Y|X) = \frac{1}{Z(X)} \exp\Big(\sum_c \sum_k \lambda_k f_k(Y_c, X, c)\Big), \tag{1}$$

where Y is the label sequence of the sentence with labels defined as either beginning of a word or continuation of one, X is the sentence of unsegmented characters, f_k is feature function, $Z(X)$ is a normalization term, and c is the index of characters in sentence being labeled. Feature functions such as character identity n-grams, morphological affix and suffix, and character reduplication features can be used in computing the probabilities of the labeled sequences. Tseng et al. and Stanford Natural Language Processing Group provide a learning framework which uses a large number of linguistic features such as character identity, morphological and character reduplication features that were extracted from the training corpora automatically thus not biased toward any particular variety of Mandarin [6] . We have tested this particular implementation on our training corpora from podcast sources and observed good performance on word segmentations.

2.2 SGNS word context learning

Once we have built a training corpus of segmented Chinese words that originates from Chinese Wikipedia

[7] and podcast sources, we can use SGNS model to learn the relationship, similarities or dissimilarities between the various words [2; 3]. Here negative sampling method is a very efficient way of deriving word embeddings: given a set of text, w, and their corresponding context, c, the model is optimizing for a maximum posterior

$$\begin{aligned} &\underset{\theta}{\operatorname{argmax}} \prod_{w,c\in D} p(c|w;\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \prod_{w,c\in D^+} p(D=1|c,w;\theta) \prod_{w,c\in D^-} p(D=0|c,w;\theta), \end{aligned} \tag{2}$$

where D, the entire set of all word and context pairs, is divided into two subsets, D^+, correct word and context pair from training corpus, and D^-, negative sample, where incorrect context is paired with words (random combinations). This objective function is to maximize, $p(D=1|w,c;\theta)$, the probability that (w,c) comes from the corpus data for D^+, and meanwhile maximize, $p(D=0|w,c;\theta) = 1 - p(D=1|w,c;\theta)$, the probability that (w,c) does not come from the corpus data for D^-. A typical choice of the probability function would be sigmoid function

$$p(D=1|c,w;\theta) = \frac{1}{1+e^{-v_c \cdot v_w}}, \tag{3}$$

where v_c and v_w are the vector representations of context and word, respectively. Putting all these together and using a logarithm form for the product of the probabilities shown in Eq. 2, we can then write the objective function for SGNS as

$$\underset{\theta}{\operatorname{argmax}} \sum_{w,c\in D^+} \log \frac{1}{1+e^{-v_c \cdot v_w}} + \sum_{w,c\in D^-} \log \frac{1}{1+e^{v_c \cdot v_w}}. \tag{4}$$

The formalism of SGNS model can be reinterpreted from a neural network perspective, where the input layer stands for the vector format of the input context words, the hidden layer is just linear propagation of a subset of this context vector representation, and output layer is based upon the inner product of the input (v_c) and output (v_w) vector. Various implementations of this three layer neural network are experimented [4; 5] to learn the vector representations of the words and context [2], both on CPU and GPU CUDA computing framework, with a rank of 400 for the latent space.

3. CONTENT BASED RECOMMENDATION MODEL

Now that the fundamental embeddings of all the words are learned, we can build an item level embedding for each individual podcast item by utilizing the available associated text features such as tags, title of the post, descriptive introduction of the post, nickname of the announcer, etc. All these text features can be indicators on the category of the audio content for example. There are a few available text features that we can utilize as indicators on what the content of the podcast item could be. These indicators include tags or keywords, item title, introductions to the item, nickname of the user who uploaded the item, and the title of the album which the item belongs to. Since not all of these text features are available for every podcast item, we just utilize the strongest indicator among all the text-based features. Table 1 shows some typical case studies on how these various text features can help indicating the item content. As you can see all these text and words provide signals or indications on the content of the podcast items, which in this example is car related discussions or news.

We can also add other additionally numerical features to this vector such as duration of the posted item, play count, download count and comment count of the item, timestamp when the item is uploaded (hour of the day, day of the week, year), and make recommendations based on a similarity metric between the historically consumed items by the user

[2] context of the words essentially also belong to the training corpus as individual words

Category of the item	Keywords or tags
汽车 (Car)	京东, 机油 (Jingdong, Engine oil,)
Category of the item	**Title**
汽车 (Car)	奔驰 E 要来了可以考虑买宝马 5 了 (Buy Mercedez Bentz E or BMW 5)
Category of the item	**Introduction**
汽车 (Car)	快过年了，现在看车的朋友很多， 但是买车的应该不多了吧。 年后开始又是一波买车高峰， 如何在年后选择一个合适的时机购入新车.... (The market of automobile before and after Chinese New Year)
Category of the item	**Nickname of uploader**
汽车 (Car)	某某说车 (Someone who talks about automobile)
Category of the item	**Album title**
汽车 (Car)	汽车立体 (Automobile stereo sound system)

Table 1: Various text features that are associated with the podcast items can provide signals to indicate the content of the item. In this example, all the five podcast items are discussing car related topics.

and items in the database. Table 2 shows some of these numeric features associated with each podcast items, where the announcer ID stands for the ID of the user who uploaded these podcast items.

4. EXPERIMENTS

4.1 Precision of the predicted category of the podcast item

We have tested these word embeddings and item embeddings using data from podcast source (Ximalaya), where a few thousand popular tracks are downloaded for testing. These popular tracks have already been manually categorized by the vendor, and these categories are used as ground truth to compare with our predictions of item similarities and dissimilarities. Figure 1 shows, after dimensionality reduction, the distribution of the podcast items in the two dimensional latent space, where different colors stand for the categories of the items provided by the podcast vendor. The cluster centroid location represent the latent embedding of the words for the categories themselves such as “music”, “audio book”, “kids”, etc. When there are actually multiple words for a category we just plot the centroid as the average of the various words for a particular category. We can see that our aggregated item embedding is actually working well in terms of clustering similar items together.

Announcer id	Track id	Duration	Play count	Upload time
1000056	19788286	360.0	108777	2016-08-10 20:22:32
1000056	19699230	318.0	6868	2016-08-08 23:42:29
1000056	20173596	370.0	19478	2016-08-19 01:22:56
1000056	19507247	360.0	12401	2016-08-04 20:27:51
1000056	19316626	450.0	50818	2016-08-01 01:48:48
1000056	18368752	570.0	16142	2016-07-12 23:03:08
1000056	18624992	360.0	56857	2016-07-18 02:56:26
1000056	17826861	634.0	70656	2016-07-01 21:05:53
1000056	18169817	230.0	76913	2016-07-09 03:01:00

Table 2: Numeric features associated with each podcast item. In this example, all podcast items are uploaded by a single user.

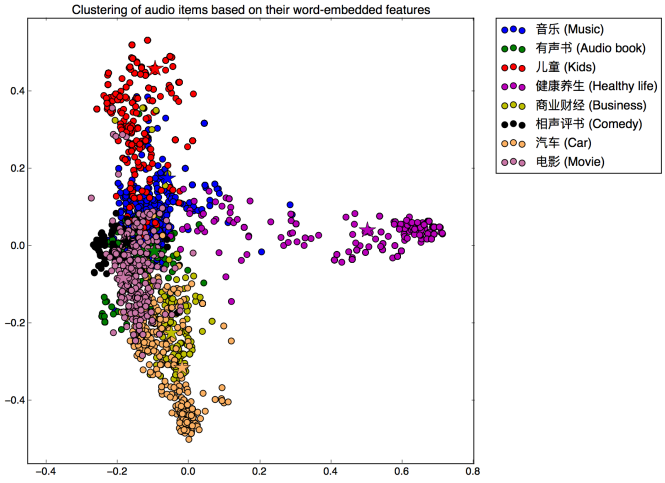

Figure 1: Clustering of the embeddings of podcast items after dimensionality reduction.

With learned embeddings for all the text-based features of the items, we can predict which category or content each item belongs to by measuring the similarity between indicators and targeted category such as “music”, “audio book”, “kids”, etc. We can then examined the precision of the predicted category or content using the cross-validation sample where la-

Indicators for prediction	CPU		GPU	
	w/o Wiki	w/ Wiki	w/o Wiki	w/ Wiki
Tags/keywords	$(84.3 \pm 0.5)\%$	$(81.9 \pm 0.6)\%$	$(72.1 \pm 0.6)\%$	$(80.4 \pm 0.6)\%$
Tags, title, introduction , nickname, album title	$(89.2 \pm 0.6)\%$	$(82.0 \pm 0.6)\%$	$(72.4 \pm 0.6)\%$	$(81.0 \pm 0.6)\%$

Table 3: Precision of the predicted category or content of the podcast items based upon various text signals.

bels of each item are already given by the podcast vendor, and the results are shown in Table 3. Here we bench marked the training process on both CPU and GPU computing infrastuctures.

4.2 Content based classification of user-item affinity

Adding the various numeric features of the podcast item, we can build a regression or decision tree based classifier to learn and then predict whether or not user will particularly like a podcast item. Since we do not have any user-item interaction data yet, here the experiment is done treating the item announcers as the users, and assuming the announcer likes a particular item if one chose to upload it to the podcast vendor. For each user we learn their affinity vector, $\vec{\theta}_u$, from their historical data of uploaded items, where each item possesses a feature vector, $\vec{x}_i$, that contains category of the item predicted by NLP, duration of the item, play count, timestamp (hour of the day, day of the week and year), etc. Cross validation of the classification can be done by predicting whether user has uploaded a particular item or not ($\vec{x}_i \cdot \vec{\theta}_u$ tells if user likes or dislikes the item). We select equal number of items for each individual user: the items that has been uploaded by this user (liked items), and also items randomly selected from a large pool that are not uploaded by this user (disliked). A Receiver Operating Characteristic (ROC) is shown in Fig. 2 by comparing predicted user-item affinity to the ground truth labels, we observe a very good Area Under Curve (AUC) in this particular test.

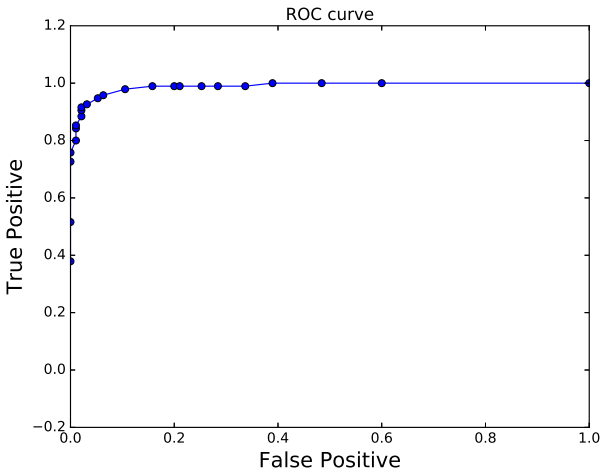

Figure 2: ROC curve for the cross validation test.

5. CONCLUSIONS

In this paper, we have proposed and examined a novel approach to utilize various text data that are associated with individual podcast item to infer or predict the content category, thus make content-based recommendations for the large volume of podcast items. Several Chinese NLP techniques are extensively used to discover similarities, relationship between fundamental Chinese words, thus provide indicator signals on the content of the item. A binary classifier is then trained on top of this predicted content category as well as several other numeric meta data features for each user using their historical data, thus make predictions on the user-item affinity. An example cross validation test is show in the experiment which indicates decent classification power of these predicted content categories.

6. REFERENCES

[1] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*, pp. 282-289, 2001.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean.

Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013.

[3] T. Mikolov and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.

[4] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50, May 2010.

[5] SPARK. http://spark.apache.org/.

[6] H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning. A conditional random field word segmenter. *Sighan Bakeoff 2005*, 2005.

[7] Wikipedia. https://zh.wikipedia.org/wiki/wikipedia.