Article Recommender System

Jayant Shelke
jayant.shelke@sjsu.edu

Mica Eldridge
mica.eldridge1@gmail.com

Slides

Abstract

We propose two novel methods for an article recommender system. In both cases, we first use unsupervised learning techniques, Doc2vec and LDA, to create a document embedding space and to automatically extract topics, respectively. We curated a Newsletter Archive dataset, in which each document is class-labeled by a user as “interesting” or “not interesting”. We convert the documents from each class to document embeddings and topic weight vectors via the Doc2vec and LDA models, and then train various classifiers using the data that results, in order to be able to predict which future articles a user will be interested in. We show that high test accuracy can be obtained using neural network architectures with a small dataset when combined with an unsupervised model.

Introduction

Many existing news article recommender systems assume a categorical approach, where an article can’t exist in more than one category at a time. This makes it difficult for someone who has inter-categorical interests to find what they’re looking for.

In this paper, we take advantage of the unsupervised learning algorithms Paragraph Vector (also known as Doc2vec) and LDA (Latent Dirichlet Allocation) to automatically produce document embeddings and detect topics, respectively.

Datasets

1. User Interests: Newsletter Archive Articles

   This dataset is used for interest modeling.

   In order to create this dataset, we gathered webpage articles that were linked from newsletter archives. We found two newsletter archives that seemed to contain sufficiently different topics. For the “interested” class, we used Daniel Miessler’s Unsupervised Learning [1] newsletter archive, which contains anything of interest to Daniel Miessler, on topics such as Machine Learning, Information Security, Technology, self-driving vehicles, and IoT. For the “not interested” class, we used the Casual Spectator Sports [2] newsletter archive, which covers topics regarding sports such as player information, game summaries, team politics, and injuries.

   For this, we wrote a program in Python [5]. First the program visits the newsletter archive page to grab the urls of the individual newsletters, then visits the newsletters to grab the urls of the referenced articles, and then downloads the articles referenced from those newsletters. In the case when the referenced article is a pdf, we parse out the full text using the Python libraries pdfminer and slate.

For the purpose of classification, we assigned a label of 1 to those articles from Unsupervised Learning and a label of 0 to those from Casual Spectator.

2. One Week Global News Feeds

   In order to train the Doc2vec and LDA models, we used the One Week of Global News Feeds dataset [4]. In total, the dataset contains 1.4 million article urls, but we downloaded and used 263,166 of the corresponding articles. In order to download the articles, we wrote a program in Python [6].

   The downloader downloads the html document at each url, parses out the plaintext body of the article using the Python Newspaper3k library, and then stores the body in a local database.

   Since we noticed many non-English articles, we filtered those out by writing a simple language detection algorithm that calculate the number of language-specific stopwords from different languages present in the documents.

3. BBC

   This dataset is used for category classification.

   The BBC Dataset [3] “Consists of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005,” including the five class labels: business, entertainment, politics, sport, tech.

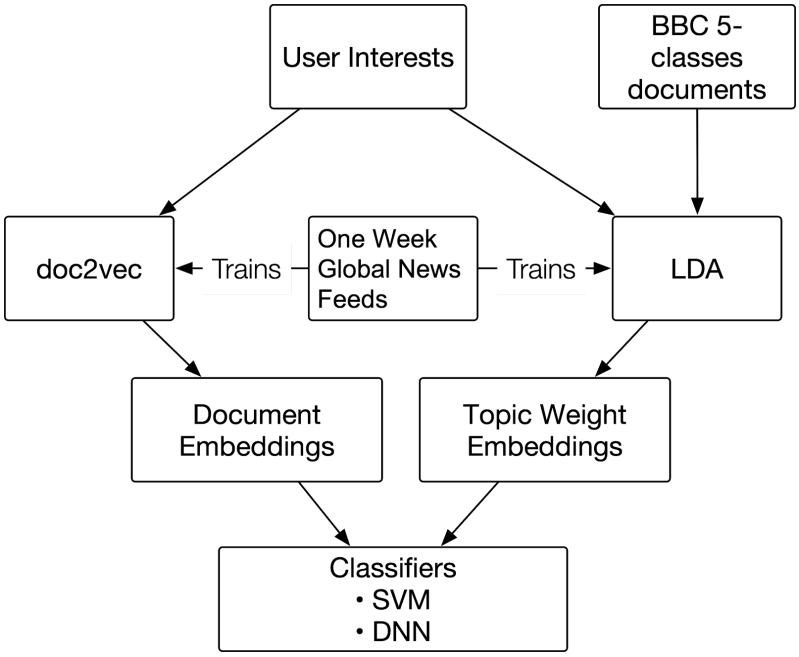
Data Cleanup:

In order to reduce noise when training the unsupervised algorithms, we cleaned the data.

Following steps were programmatically accomplished for creating meaningful data that would help to create good models.

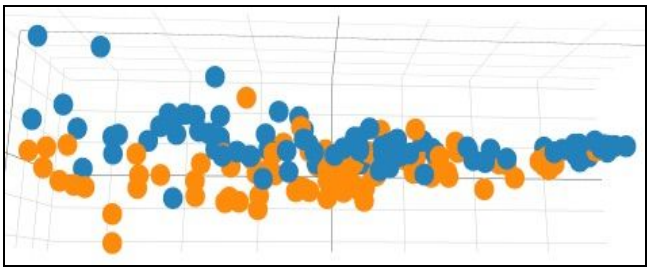
- Ignored articles with languages other than English:
  - There were no clear and simple libraries for language detection
  - We used stopword frequency counts from various languages to try to automatically detect the language of each document
- Removed stopwords, articles, punctuation, numerical values, and additional features whose removal would not change the topical meaning of the documents

Architecture:



Doc2vec: A machine learning algorithm that creates document-level fixed-length float-entry vectors for each document. Similar documents will have closer embedding vectors than dissimilar documents. We chose a vector length of 100 entries.

PCA: A machine learning algorithm that enables dimensionality reduction by trying to compress important information into less space by using weighted combinations of the original features. For the purpose of analyzing our trained Doc2vec model, we used PCA to reduce the dimensionality of our user interest dataset from 100 dimensions to 3 for the purpose of creating a 3D plot and viewing the separation.



LDA: A machine learning algorithm based on "*Latent Dirichlet Allocation*" which creates topic weight vectors useful for topic modeling based on documents. It internally used word and document vector understandings to do so while updating the corpus with each document that it has trained. The output appears as topics and their corresponding percentages, which should add up to approximately 1. In order to pass these to classification machine learning algorithms, we converted them to fixed-length vectors by putting document's corresponding percentages of the same topic in the same column, and placing zeros in all entries that documents were detected as not containing.

Support Vector Classifiers: A class of machine learning algorithms that are very popular for classification problems.

Deep Neural Networks: A deep neural network for solving binary and categorical classification problems.

Experiment 1:
The first experiment explores the problem of figuring out articles with multiple topic interests being mingled together. This was handled with an approach of interest modeling instead of topic modeling, as topics were not explicitly determined or detected, but rather detected indirectly through the document embedding space. We decided to model the user interest based on dataset 1, which is detailed above. Based on a user's past upvotes and downvotes, the model can predict whether the user will be interested in a new article.

While embedding spaces capture semantic meaning between vectors, they are difficult to understand because we usually can't explain what the model actually learned. For that reason, we decided to try a classification problem using a model that gives human-understandable results.

Experiment 2:
The second experiment considers news article category classification. First we trained an LDA that can automatically detect topics. Then we used the LDA to detect topics of documents with known categories, and then train models to detect the category based on the LDA-determined topics of new documents.

Model implementation details:
Doc2vec: In this intermediate model, during the training phase, the input is an iterator of variable-length lists of strings, where each list corresponds to a document, and the strings in that list are the tokens from the document, in the order they appeared in the document. In the prediction phase, the input follows the same input pattern as during the training phase, and the output is a fixed-length embedding (100 in our case) with float entries that typically range from -3 to 3.
LDA: In this intermediate model, the input is similar to Doc2vec model except that it is a series of documents where each document is represented with tokens from the document. These tokens are transformed into id-token pairs. These pairs are then used to create a corpus, where each id of the token has a floating point value. With more and more training and more text in the corpus, the LDA model is able to learn word vectors, document vectors and topic vectors all together. In the testing/validation/prediction phase, the input is a tokenized document and the output is vector of topic mixture.
Sample output: [Topic1: 68%, Topic2: 27%, Topic3 : 5%]
Keras DNN model:
When testing the input from vector models, the Deep Neural Network was defined as follows:
<u>Doc2vec binary classification:</u>

- Neurons in each layer : (10, 30, 30)
- Activation Functions in each layer: (relu, sigmoid, sigmoid)
- Output: 1 neuron with sigmoid activation
- Optimizer: Adagrad
- Loss: Binary cross entropy

<u>LDA2vec 5-class classification:</u>

- Neurons in each layer : (10, 10, 10)
- Activation Functions in each layer: (relu, relu, relu)
- Output: 5 neurons with softmax activation.
- Optimizer: rmsprop
- Loss: Categorical cross entropy

Results :

Experiment 1:
First we trained a Doc2vec model on 229,410 unlabeled documents from the One Week Global News Feeds dataset and the user interests dataset. We included the user interests dataset to ensure that the Doc2vec model had the proper vocabulary to be able to model those documents correctly. The input to Doc2vec was an iterator of variable-length lists, where each list corresponds to a document and contains the individual string tokens of that document in the order they appear in the document. In order to prevent reading the entire 229,410 into memory at once, we trained in an online fashion from a local Postgres database, fetching each document as it was needed. We trained for 19 epochs, each epoch taking approximately 2.5 minutes, which took a total of approximately 47.5 minutes.

Once we had the trained Doc2vec model, we chose a small dataset of 90 articles from each of the positive and negative classes of the user interests set, and split 20% of each class into a test set. We then used a list of Scikit-Learn Support Vector Classifiers. The train/test positive/negative dataset breakdown and train/test accuracies for each Support Vector Classifier are shown below. The value for nu in the NuSVC was taken as the default 0.5.

	Train	Test	Model	Train Accuracy	Test Accuracy
Label			**LinearSVC**	0.993056	0.805556
Positive	72	18	**SVC**	0.812500	0.833333
Negative	72	18	**NuSVC**	0.951389	0.861111

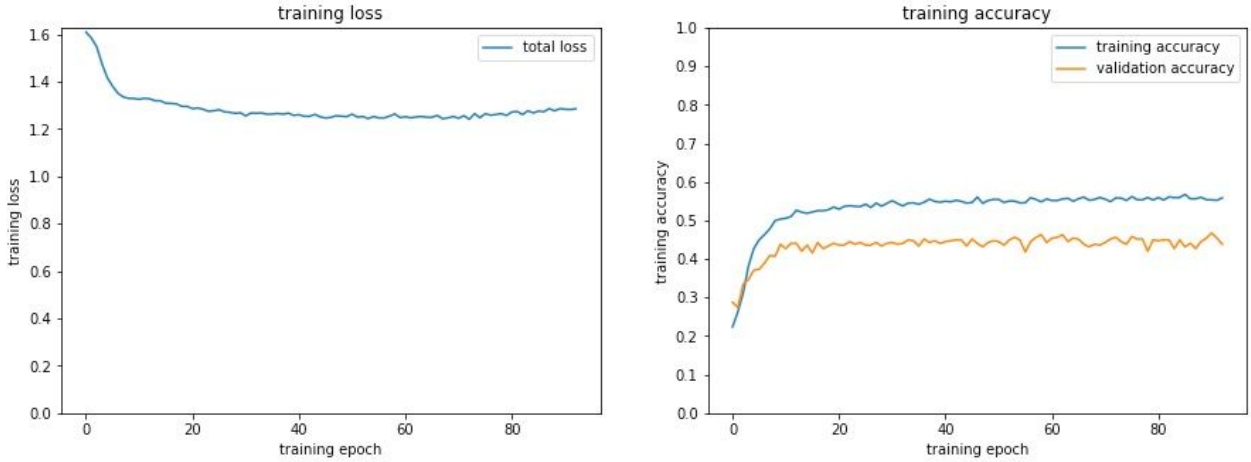
SVC accuracies on User Interest classification, without Cross Validation

We also used a Keras Deep Neural Network for binary classification. The architecture is described above. The corresponding training and testing accuracies, without cross validation, were:

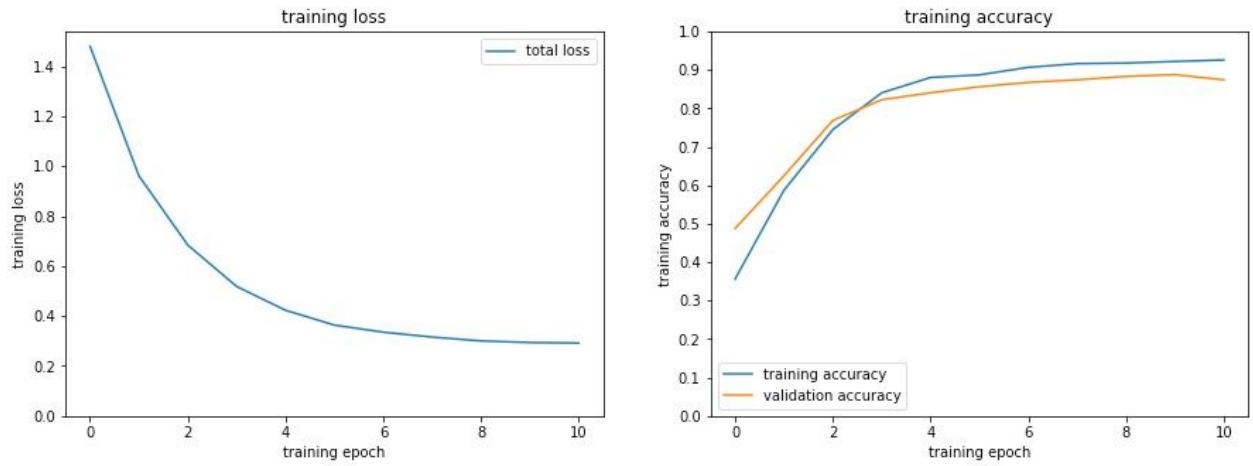
- Training Accuracy: 0.854
- Validation Accuracy: 0.889

Experiment 2:
We trained the LDA model on the unlabeled One Week Global News Feeds dataset. The first time we initialized an LDA model, we instantiated it with 31,000 topics. This means that the

model will try to model 31,000 topics during training. Once the LDA had been trained, we passed the labeled BBC dataset (5 classes) through the model to obtain topic weight vectors, which we then converted to fixed-length vectors, and then tried training Keras deep neural networks on those vectors. We found that trying to model 31,000 topics caused the model to not be able to learn very well, as the validation accuracy capped at about 40%. We re-instantiated an LDA with 1,000 topics and re-trained from scratch, and performed the same process on the BBC dataset with this new model, and were able to obtain 88.8% validation accuracy on 5-class classification. The epochs versus training and validation accuracy graphs are shown below for the 31,000- and 1,000-topic models.



5-class classification on 31,000-topic LDA topic weight vectors



5-class classification on 1,000-topic LDA topic weight vectors

As shown by the plot above, the model performed well when the model was tuned to learn about 1,000 topics from the base corpus built out of roughly 200,000 documents. The data used for testing/validation after the corpus learning was the BBC dataset which had 5 classifications for 2225 documents.

	Total	Train	Test	Percent test
Business	510	408	102	0.2
Sport	511	409	102	0.2
Politics	417	333	84	0.2
Entertainment	386	309	77	0.2
Tech	401	321	80	0.2

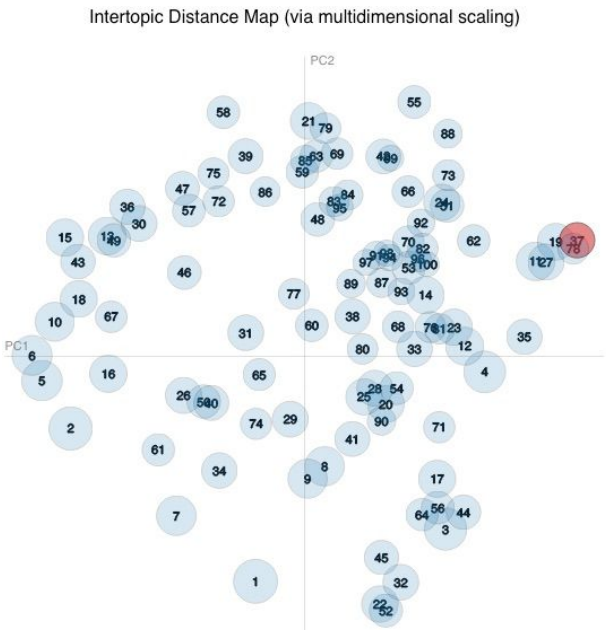
Amount of data used for training and testing LDA on 5-class classification using Keras Deep Neural Networks

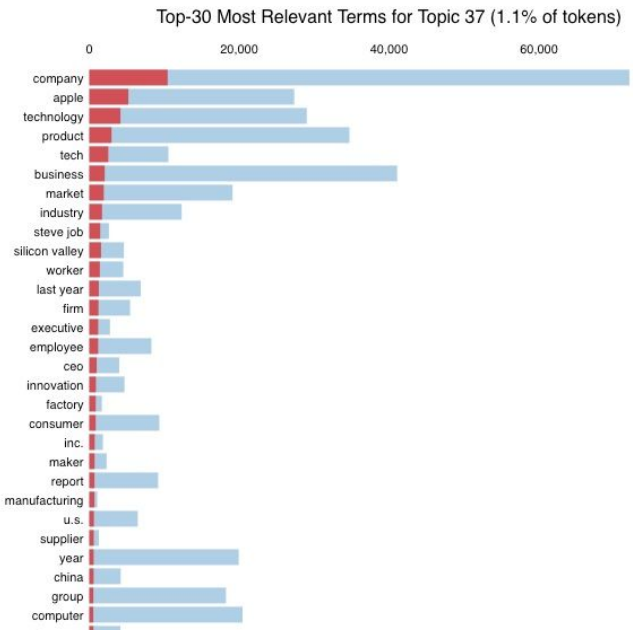
We also tried the LDA model on the User Interests dataset using 5-fold Cross Validation, and obtained the following test accuracies (mean +/- standard deviation):

```
LinearSVC   : 0.95 (+/- 0.004)
OneClassSVM : 0.77 (+/- 0.183)
NuSVC nu=0.1: 0.97 (+/- 0.006)
NuSVC nu=0.2: 0.94 (+/- 0.009)
NuSVC nu=0.3: 0.91 (+/- 0.010)
Keras       : 0.93 (+/- 0.042)
```

We also tried the LDA on 5-class classification dataset with cross validation on the Keras model, and obtained the testing accuracy of 0.924±0.017.

Learnings - The best accuracy to be obtained was about 88%.
The reason that we had better understanding with 1,000 topics was because the model seemed to understand just enough about the topics to make reasonably accurate distinctions. With 31,000 topics, it was not able to distinguish noise from signal and so produced garbage results. When trained with 100 topics, the model understanding was too constrained and could not generalize well for the 5 classifications.
This can be seen from this diagram which is a visualization of the internal learning of LDA2vec model





Conclusions:

From the above experiments we realized that,

1. User interest modeling could be a really helpful way of solving the problem of news article recommendation with intermingling interests.
2. Topic modeling done via unsupervised learning though is not directly useful, but, can act as a very good supportive tool in learning the latent understandings of a document which can be trained via other models to accurately work with topic modeling.
3. Unsupervised learning can have benefits in making supervised models work better.
4. Unsupervised learning requires a large amount of data for training.
5. Distortion towards learning contextual meanings of texts can be eliminated by curating the dataset with proper cleaning routines.

References

1. Unsupervised Learning, Daniel Miessler, https://us8.campaign-archive.com/home/?u=6a9e465ab1570df8aaecb2292&id=49fdb7d723
2. Casual Spectator Sports, Casual Spectator, https://us9.campaign-archive.com/home/?u=fd29bb4bf60ba78d658031aba&id=2e9018b243
3. Dataset: BBC, “Consists of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005.”, D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006., http://mlg.ucd.ie/datasets/bbc.html
4. One Week of Global News Feeds, 1.4 million article urls, Rohit Kulkarni, https://www.kaggle.com/therohk/global-news-week

5. Chromatic News Newsletter Archive Downloader, Mica Eldridge, https://github.com/mica5/chromatic_news/tree/master/download_newsletter_archives
6. One Week of Global News Feeds data downloader, Jayant Shelke, https://github.com/LearningOwl/ArticleDownloaderAndParser