Research Paper Recommender System Based on Deep Text Comprehension

Dongyu Ru Kun Chen

SJTU

May 27, 2018

Table of Contents

Introduction

A recommender system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.

Introduction

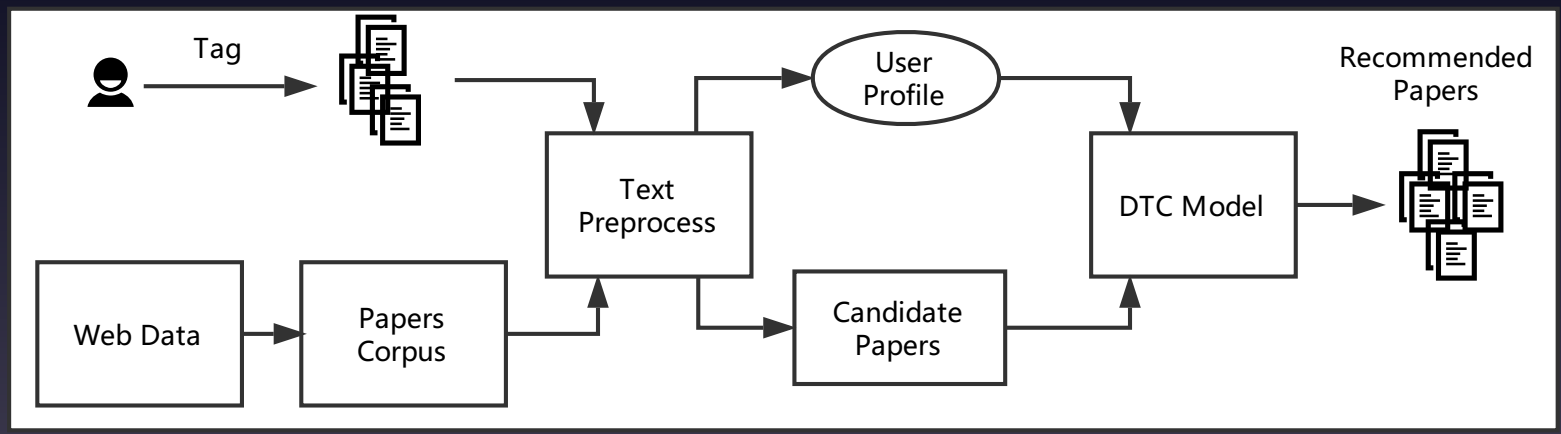
Many recommendation classes have been utilized over the past few years, among which, typically, the following two classes are most popular.

- Content-Based Filtering
- Collaborative Filtering

Introduction

- Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties.
- Collaborative filtering approaches build a model from a user's past behaviour as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in.

Framework


Tag
User Profile
Recommended Papers
Text Preprocess
DTC Model
Web Data
Papers Corpus
Candidate Papers

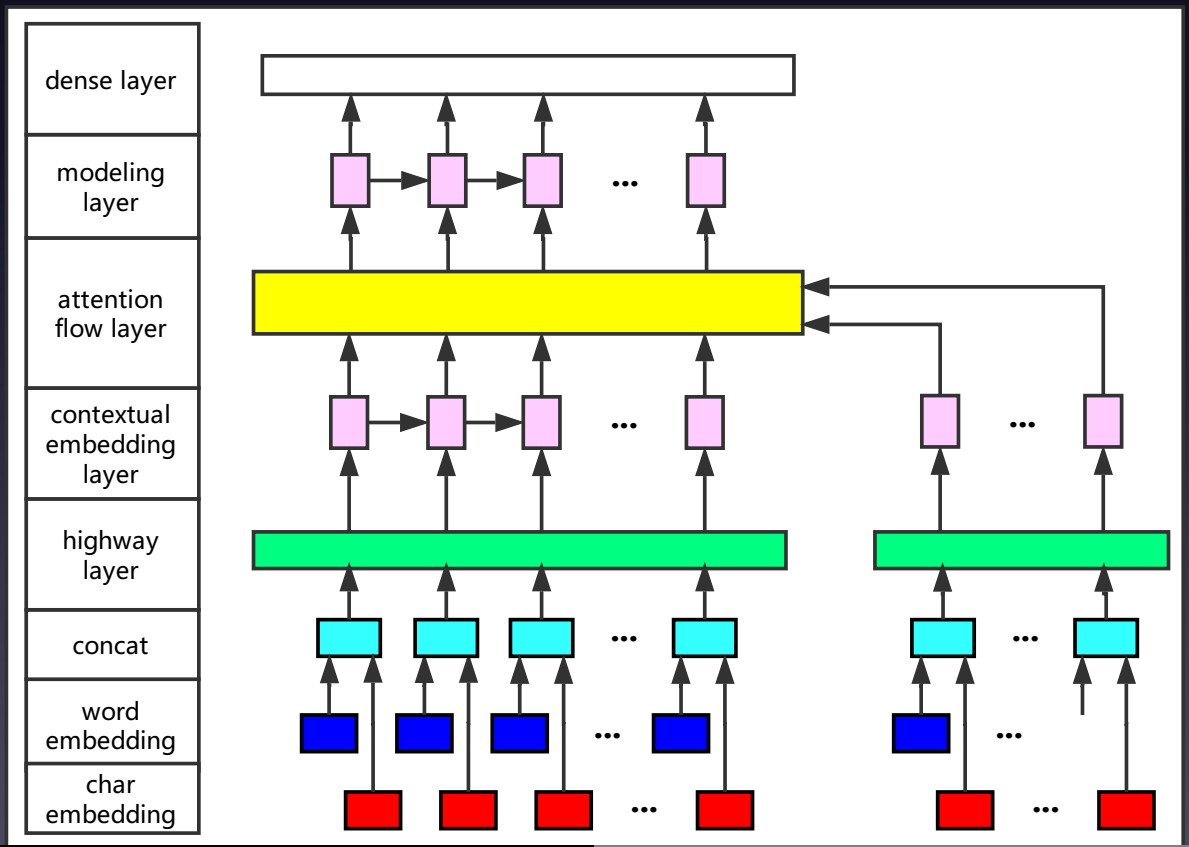
Framework

We came up with the framework above based on a typical Content-Based Filtering Model. The main difference is that, we replace the original matching model in the CBF system with our DTC Model. Because we claim that our Deep Text Comprehension model has higher capacity to recognize the patterns of given text than simple n-gram or TF-IDF based models.

Framework

We think the rest parts except for the DTC model are relatively mature and well exploited. So we focus on the DTC model, which is actually a deep neural network.

Model


dense layer
modeling layer
attention flow layer
contextual embedding layer
highway layer
concat
word embedding
char embedding

Model

The DTC model is a deep LSTM-based neural network which consists of mainly 7 layers, as shown in Figure above. It takes as input the words and characters of the paper text. And output a similarity score between the input papers. The detail structures are introduced in the following part.

Model

- **Character Embedding Layer** This layer maps each word to a vector space using character-level CNN (Convolution Neural Network). Let $\mathbf{a} = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_T\}$ and $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_T\}$ represent the input words of two papers. Characters are embedded into vectors, as 1D inputs to the CNN, whose size is the input channel size of CNN. The outputs of CNN are max-pooled over the entire width to obtain a fixed-size vector for each word.

Model

- **Word Embedding Layer** This layer maps each word to a high-dimensional vector space. Pretrained word vectors, GloVe, are used to obtain the fixed word embedding of each word. The output of Word Embedding Layer and Char Embedding layer are concatenated together as representation of input text.

Model

- **Highway Layer** This layer takes as input the concatenation of two sequences of embedding vectors in word-level. And it performs as a gate to leak part of original information of input directly to next layer. Let **x** represent the input.

$$
\begin{aligned}
T(x) &= \sigma(W_T x + b_T) \\
o(x) &= relu(W_o x + b_o) \\
O(x) &= T(x) \cdot o(x) + (1 - T(x)) \cdot x
\end{aligned}
\tag{1}
$$

Model

- **Contextual Embedding Layer** In this layer, a LSTM(Long Short Term Memory) Network is applied after the Highway layer output. The output states of LSTM are concatenated and transmitted to the next layer. Till now, feature representation on different granularity has been obtained.

$$y_t = BiLSTM(y_{t-1}, x_t) \tag{2}$$

Model

- **Attention Flow Layer** Here, contextual embedding output of two papers are input to the Attention Flow Layer to get a mutual-aware representation of input papers.

Model

- **Modeling Layer** The Modeling Layer are constructed by another LSTM layer. The input of modeling layer is attention output stacks. It captures the interaction in the mutual-aware representation of input papers.

Model

- **Dense Layer** The Dense Layer acts as the output layer of this model, which takes the final state of Modeling Layer as input, use a fully-connected layer and sigmoid function to get score of similarity.

$$score = sigmoid(W_s^T M) \tag{3}$$

Baseline

There are two baselines selected to compare with our Model on matching performance.

- TF-IDF
- Simhash

Baseline

- TF-IDF, short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
- Simhash is a technique for quickly estimating how similar two sets are. The algorithm is used by the Google Crawler to find near duplicate pages.

Baseline

Some important formulas of TF-IDF:

- $tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$
- $idf_i = log \frac{|D|}{1+|j:t_i \in d_j|}$
- $tfidf(i,j,D) = tf_{i,j} * idf_i$

Baseline

Main procedure of simhash:

- Default hashsize B = 64, let V = [0] * B
- Break the phrase up into features, and hash each feature using a normal 64-bit hashing algorithm
- For each hash, if bit_i is set then add 1 to V[i], else take 1 from V[i]
- simhash bit_i is 1 if $V[i] > 0$ and 0 otherwise
- Sort all hash values and check adjacent, then rotate 1 bit, repeat for B times

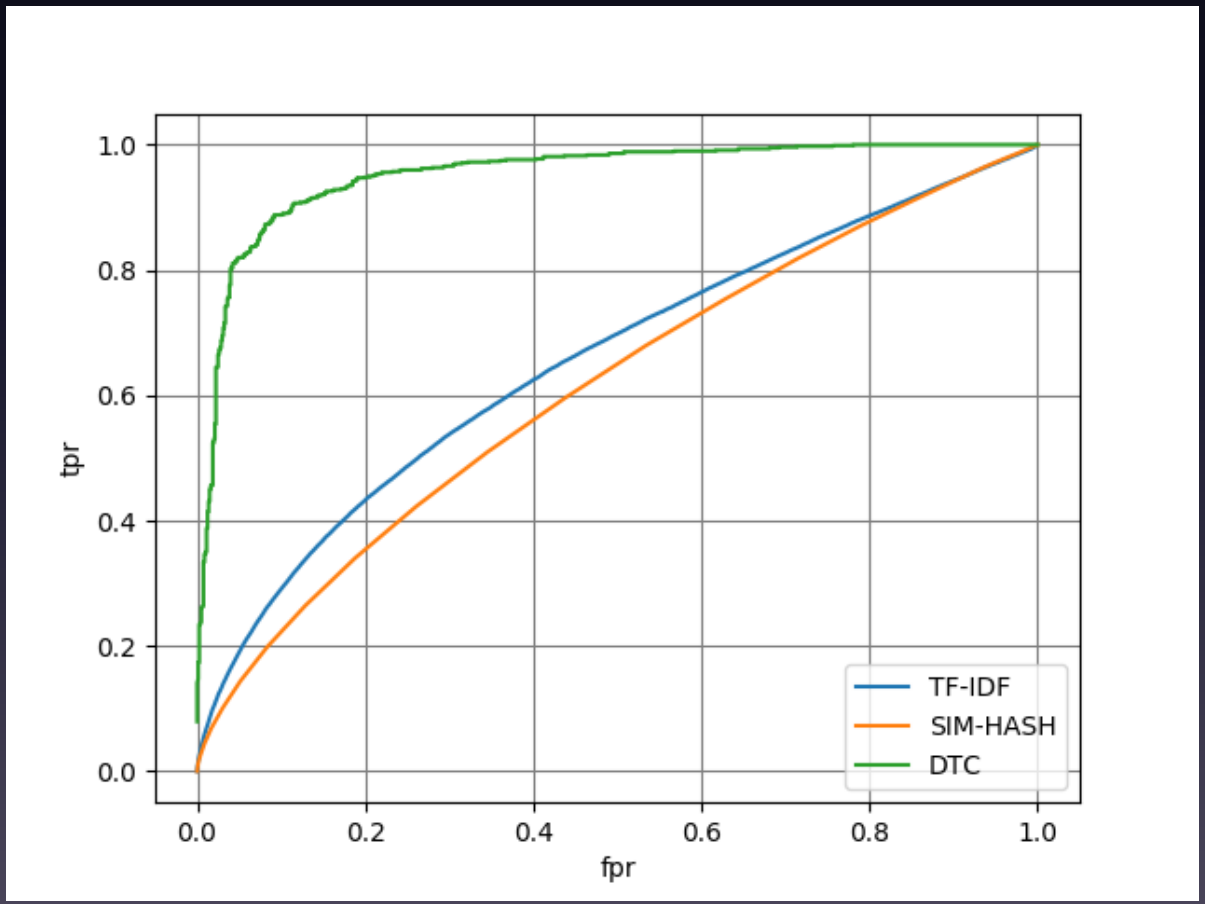
Experiments

To prove our model performs better to play as a matching model in Research Paper Recommender System. We collect a dataset to verify the performance of our model and baselines. Restricted by the limited computation power, we randomly selected 1M papers from the dataset for validation. After filtering out bad cases in the dataset. Finally we perform the experiments on a dataset of 200K papers.

Experiments

30% of the datasets are reserved as test set. And experiments on baselines are directly performed on test set without training. We evaluate our DTC (Deep Text Comprehension) Model with ROC (Receiver Operating Characteristic Curve) as shown in following Figure and AUC (Area Under Curve) as shown in following Table.

Experiments



Experiments

Table: AUC comparison of matching performance

	TF_IDF	SIM-HASH	DTC
AUC	0.65	0.61	**0.95**

Experiments

One more thing need to mention is that our model runs slower than those 2 baselines. For a same dataset with 60K data items, TF-IDF runs for 3 mins, while Simhash runs for 3 hours, both on CPU i5-5200U. Our model needs about 10 hours, and extra GPU support needed.

Conclusion

We proposed a Deep Text Comprehension based Recommender System, which replace the original matching model in a CBF Research Paper Recommender System with a Deep Neural Network. And we claim the DTC model has higher capacity to recognize the patterns in text. Experiments indicate that our DTC model performs better than two baselines mentioned in the report. However, the extra running time and computing power is still a problem to be fixed.

References

- Beel J, Gipp B, Langer S, et al. paper recommender systems: a literature survey[J]. International Journal on Digital Libraries, 2016, 17(4): 305-338.
- Ferrara F, Pudota N, Tasso C. A keyphrase-based paper recommender system[C]//Italian Research Conference on Digital Libraries. Springer, Berlin, Heidelberg, 2011: 14-25.
- Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
- Kim Y, Jernite Y, Sontag D, et al. Character-Aware Neural Language Models[C]//AAAI. 2016: 2741-2749.
- Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.
- Srivastava R K, Greff K, Schmidhuber J. Highway networks[J]. arXiv preprint arXiv:1505.00387, 2015.
- Seo M, Kembhavi A, Farhadi A, et al. Bidirectional attention flow for machine comprehension[J]. arXiv preprint arXiv:1611.01603, 2016.

Questions

Thank you for your time!