Course Project Report (Jul-Dec 2020)

Deep Learning-based Online Alternative Product Recommendations at Scale

Submitted By

Amodh Shenoy (171IT108)
Sagar G (171IT136)
Akhil G (171IT106)

as part of the requirements of the course

Information Retrieval (IT458)

under the guidance of

Dr. Sowmya Kamath S, Dept of IT, NITK Surathkal

undergone at

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

NOVEMBER 2020

Department of Information Technology
National Institute of Technology Karnataka, Surathkal

C E R T I F I C A T E

This is to certify that the Course project Work Report entitled **"Deep Learning-based Online Alternative Product Recommendations at Scale"** is submitted by the group mentioned below -

Details of Project Group

Name of the Student	Register No.	Signature with Date
Amodh Shenoy	171IT108	
Sagar G	171IT136	
Akhil G	171IT106	

as the record of the work carried out by them as part of the course **Information Retrieval (IT458)** during the semester **Jul - Dec 2020**. It is accepted as the Course Project Report submission in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Information Technology.**

(Name and Signature of Course Instructor)
Dr. Sowmya Kamath S

D E C L A R A T I O N

We hereby declare that the project work report entitled **"Deep Learning-based Online Alternative Product Recommendations at Scale"** submitted by us for the course **Information Retrieval (IT458)** during the semester **July - Dec 2020**, as part of the partial course requirements for the award of the degree of Bachelor of Technology in Information Technology at NITK Surathkal is our original work. We declare that the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles elsewhere.

Details of Project Group

Name of the Student	Register No.	Signature with Date
1. Amodh Shenoy	171IT108	
2. Sagar G	171IT136	
3. Akhil G	171IT106	

Place: NITK, Surathkal
Date:

IR Project – Final report submission format

Amodh Shenoy[1], Akhil G.[1], Sagar G.[1], and Dr. Sowmya Kamath [2]

Abstract— **Ecommerce companies are utilising all the technology available to them to maximise their profits in anyway possible. One such category that they have mad extensive use of is recommender systems. These models are built to support the customer make the right decision by suggesting products which the customer might feel useful. These systems try their best to fit the customer's needs by trying to profile the customer's habits by tracking every single interaction the customer has with their interface. This gives rise to the Cold Start Problem, where initially due to lack of any data, the system does not know what to suggest to the customer and suggests generic items which might or might not fit the customer's demands. To tackle this issue, we can utilise textual product information (e.g.product titles and descriptions) as well to build relations between products. In this project we do exactly the same, and combine this with customer behavior data to recommend alternative products. In order to better capture the semantic meaning of product information, we build two different LSTM models, one which works with product description and the other which works with customer behaviour data, and fleece their outputs together to give the customer a very well fitted set of product suggestions.**

I. INTRODUCTION

Partially because of it's extensive utilization by the ecommerce industry, recommender system technology is being researched on heavily and these systems are improving by leaps and bounds as the days go by. The main goal of this technology is to conduct extensive analysis on the customer behaviour, the way the interaction happens with the interface, as well as the product sales history and return back a very tailor fit product recommendation to the customer.Traditional methods used for this task include content-based recommendation, collaborative filtering as well as Hybrid recommendations. All these methods have been analyzed thoroughly by assessing the accuracy of their recommendations. Although most of these techniques do satisfy their purpose for customers which are active on the platform or for products which have a decently sized and already established customer base, they fail to give accurate results when faced in situations where a new customer signs up on the platform or if new products are made available for sale. This is called the Cold Start Problem. Our aim is to come up with an accurate solution for the same which is also efficient and scalable at the same time. The main idea is to combine two very traditional methods, one being utilising customer behaviour and the other utilising product's textual data available, by fleecing the final outputs from both these methods.

[1]Amodh Shenoy, Akhil G., and Sagar G. are final year Bachelor of Technology students in the department of Information Technology at NITK Surathkal, Karnataka, India.

[2]Dr. Sowmya Kamath is an Assitant Professor(Grade I) in the department of Information Technology at NITK Surathkal, Karnataka, India.

II. RELATED WORK

(Doha et al., 2019)'s research has been centred around the drawing consumers towards social commerce primarily for social values. The addressed research model of the works has been utilizing the three perspectives such as economic,utilitarian, and social perspective. The method used here was majorly involving the consumer's behavior data, i.e., how he/she interacts with the interface, the number of views, clicks and such.

(Osadchiy et al., 2018) have based their entire research around content-based suggestion models and collaborative filtering. They have worked extensively on building and perfecting food recommender systems to be specific. They implemented recommender algorithms based upon an inherent association rule, social graph, and analyzing pairwise association. The highlight of their research were the pairwise association rules (PAR), which help in building pairs of food items which can be grouped together because of the cosine similarity in their content, which includes the ingredients, recipes, cuisine and such detailed data.

(Strub et al., 2016). They designed a system of denoising autoencoder-based recommender system (UT CDAE) that model's user rating trend when detecting the top-N recommendation of a user. UT CDAE technique produced outperformed other denoising auto-encoder based techniques DAE [19] and CDAE for the comparatively lower value of ICR(0.2 and 0.5). A discernible proportion of users' rating-trend was maintained by low ICR values in the data set and therefore offer useful information to UTCDAE to model users' rating behavior. There are three kinds of rating methods that have been considered in the hyperparameter values such as learning rate, regularization rate 1 and regularization rate 2. They compared the result with CDAE and DAE in accordance with the terms and conditions of several metrics showing the improvement and reliability of the UT CDAE method.

(Hernández et al., 2017) performed a complete analysis of the user interface interaction on an e-commerce platform and tracked metrics like visiting series, frequency and, time spent on each section of the website. They did this through the recorded clickstream data on the platform on the internet. They then went ahead to group the behaviours patterns of multiple customers and develop a common pattern among the same. This helped them find consumer similarity and create homogeneous clusters by selecting random objects as initial leaders of the groups and finally detect concrete customer behaviour patterns.

The traditional method for recommender systems is content-based recommendations (Lops et al., 2011). This method can handle the cold start problem well. Collaborative Filtering is another method based on user behaviors. For example,Matrix Factorization (Koren et al., 2009) is a widely used method for collaborative filtering. Of our two baseline algorithms, one is considered as content-based and the other utilises customer behaviour(customer purchase history to be specific). Deep learning now has been widely used not only in the academic community, but also in industrial recommender system settings, such as Airbnb's listing recommendations (Grbovic and Cheng, 2018) and Pinterest's recommendation engine (Ying et al., 2018). We shall now proceed to outline our plan of action to achieve the goals outlined so far.

III. Proposed Methodology

The first thing that has to be done is process the data so that it is ready to be fed to the model. We have two datasets for the two models, both from the Santander Product Recommendation Challenge on Kaggle. The first dataset is meant for the model which is supposed to train on customer shopping history. The second dataset is supposed to be for the model which is supposed to train on the product descriptions. Both of these datasets are in Spanish language. However, this does not pose to be an issue to us since the content-based model trains upon the individual tokens generated, and the dataset for the history based model is converted to a one-hot-encoding format so essentially breaking the language barrier. Once the models are trained, we will be merging the output from these into a Siamese Network. We will also be having a middleware layer which takes input from the user, feeds it to both these models and merges the output and presents it to the user. In the next section we will be going into the exact details of this.

IV. Implementation Specifics

As mentioned, the first stage will be preparing the data for training. For the customer history data, we use the Santander Product Recommendation Challenge's dataset and for the second dataset, we extract the product ID, product title and description as the raw textual data from product descriptions provided through the same dataset. For the product history dataset, the majority of the preprocessing involves converting the dataset into a one-hot-encoded format and for the textual-description dataset, we first tokeninze the words and map them to a number which denotes the rank of the words when sorted in a decreasing order of frequency.

We then begin training our models. The entire core of our models is the Bidirectional LSTM. Bidirectional LSTMs are variants of the simple RNN cell that is capable of learning long-term dependencies. The main dependencies we are trying to detect here are the relations between the words in the textual description for the products and the relations between two different products bought by the customer on the same order. LSTMs are the perfect models for this specific purpose, and building up on these are the bidirectional LSTMs which are two LSTMs coupled together to train on the data in the right order as well as the reverse order, this way retaining the relationship regardless of the order.

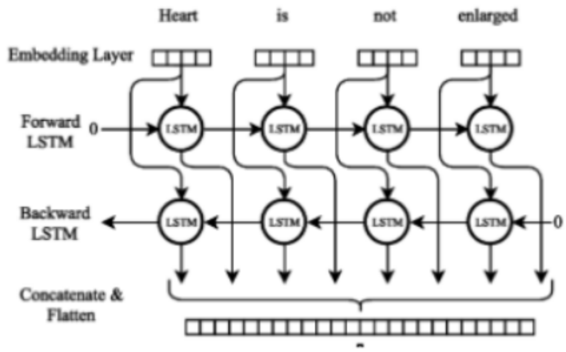

Fig. 1: Bidirectional LSTM

We combine the two bidirectional LSTM components to make a Siamese Network which can take the data to be trained and create embeddings of the products. The space for the product embeddings best captures the semanticized sense of knowledge and consumer desires. Text content is in a sequential format and network order is essential for the network. In order to represent the correct association among the words, we need to consider the order of words in both directions, the correct order and the reversed order as well. For this we use bidirectional LSTMs, which do exactly the same. When working with textual data, it is entirely possible that the output is also dependent on the future outputs as well. This is especially true for applications where the attributes of the word or phrase we are trying to predict may be dependent on the context given by the entire enclosing sentence, not just the words that came before it. In our case, this is a reference to how the product description is phrased, and also how the product history entries are coupled. This problem can be solved using a bidirectional LSTM, which are essentially two RNNs stacked on top of each other, one reading the input from left to right, and the other reading the input from the right to the left. The output at each time step will be based on the hidden state of both RNNs. Bidirectional RNNs allow the network to place equal emphasis on the beginning and end of the sequence, and typically results in performance improvements. In 2 we can see the architecture of the final product. We create and build the network using Keras with TensorFlow. As an optimizer, we pick RMSprop. Binary cross entropy is the loss function.

It takes approximately 20 hours to converge on Siamese Network preparation. The next move is to load the weight of the best model for making product incorporations. The next step is that we remove the last cosine similarity layer and the second input branch which processes the second product of the product pairs. We use the Bidirectional LSTM and the Embedding layer only. The final implication is that the product title's secret condition and the product definition are concatenated.

Finally we fleece the outputs of the two models. Since our content-based model is more accurate than the other, we give more weightage to it's output while fleecing. Fleecing implies intertwining the outputs of the two models, keeping the common items and intertwining the rest of them.

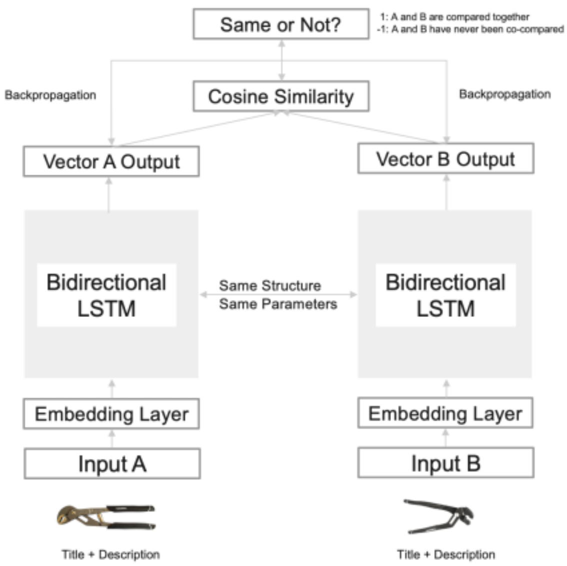

Fig. 2: Siamese network of bidirectional LSTM

V. Experimental Results and Analysis

For the product descriptions model, embedding items are generated on the basis of product names and descriptions. The job is to measure the distances for the remaining millions of product embeddings using similarity metrics, such as cosine similitude, and to distinguish similarity effects between the higher and the lower to achieve the recommended top 5 products.

For the customer history model, the guidelines are listed according to the co-comparison counts. Because of the problem of cold start, many items in the catalogue don't even allow labels in the reference results.

We select 0.8 as the cutting threshold for the cosine similitude score for Deep Learning Based. After analysing thousands of random sampled anchors from our catalogue data and suggestions by the original authors of the base paper, the threshold is selected and validated according to the test data that was provided in the Santander Product Recommendation dataset. We only hold suggestions that are close to each anchor product at least 0.8. Table 1 displays the output from the three techniques.

	Accuracy
Based on description	94.85%
Based on history	86.19%
Combining the two	92.77%

TABLE I: Results

VI. Conclusion and Future work

Recommend schemes are essential revenue-enhancing features for online retailers. We build a comprehensive learning method for developing embeddings based on a Siamese Network with Bidirectional LSTM in order to help clients identify alternative products automatically. We took collected and compared data from the client clickstream and product text data to train the network and create embedding space. To help clients find alternative products quickly. Our methodology increases substantially the coverage and reminder and accuracy of related items. To improve on the current product, we can try to implement other models in place of LSTMs to see if we achieve any improvement in the rsults. We can try to replace the bidirectional LSTM with XGBoost and evan try to incorporate collaborative filtering as well. The data too on the other hand can be filtered out extensively to achieve almost zero duplication of entries and avoid training on redundant data. With all these above changes, we can achieve a higher success rate.

VII. Individual contributions

- Cleaning and preprocessing of dataset : Sagar G
- Training LSTMs on Santander Product Recommendation Kaggle competition dataset and the Amazon product description dataset : Akhil G
- Training LSTMs on the proposed improvement: Amodh Shenoy

REFERENCES

Doha, A., Elnahla, N., and McShane, L. (2019). Social commerce as social networking. *Journal of Retailing and Consumer Services*, 47:307–321.

Grbovic, M. and Cheng, H. (2018). Real-time personalization using embeddings for search ranking at airbnb. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*.

Hernández, S., Álvarez, P., Fabra, J., and Ezpeleta, J. (2017). Analysis of users' behavior in structured e-commerce websites. *IEEE Access*, PP:1–1.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42:30–37.

Lops, P., de Gemmis, M., and Semeraro, G. (2011). *Content-based Recommender Systems: State of the Art and Trends*, pages 73–105.

Osadchiy, T., Poliakov, I., Olivier, P., Rowland, M., and Foster, E. (2018). Recommender system based on pairwise association rules. *Expert Systems with Applications*, 115.

Strub, F., Gaudel, R., and Mary, J. (2016). Hybrid recommender system based on autoencoders. pages 11–16.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*.