Aspect based Neural Recommender using Adaptive Prediction

Avinash Bhojwani
Department of Information Technology
National Institute of Technology Karnataka
Surathkal
avinashbhojwani.201it211@nitk.edu.in

Vanshika Jolly
Department of Information Technology
National Institute of Technology Karnataka
Surathkal
vanshikajolly.201it263@nitk.edu.in

Saksham Goel
Department of Information Technology
National Institute of Technology Karnataka
Surathkal
sakshamgoel.201it152@nitk.edu.in

Anand Kumar M
Department of Information Technology
National Institute of Technology Karnataka
Surathkal
m_anandkumar@nitk.edu.in

***Abstract*—Recommendation systems are information processing systems that analyze user behavior and make suggestions relevant to the users' interests. These systems recommend movies, products etc., based on many different factors. These recommended items are the items which are most likely in the interest of the users. This work focuses on aspects of users' textual comments to make recommendations that are relevant to users. We propose a model that recommends the items based on extracted aspects, taking the reviews and information from the items. Based on the item's aspect level importance, a user may rate them high or low, and we have modelled our recommendation system considering the same. This calculation is based on the neural co-attention mechanism. The proposed model points to various shortcomings of the model that were introduced before. We have used the datasets from amazon, which are extracted using web scraping. The existing recommendation system predicts ratings alone, while the proposed model gives rankings and ratings. The model's efficiency is checked using precision, recall and F1 score.**

***Index Terms*—Aspect- Based Recommendation System, aspects, Neural co-attention, amazon datasets, web-scraping.**

I. Introduction

A recommendation system is an information retrieval system that predicts the rating or preference a user would like to give to an item of his/her preference. Sometimes people need clarification on recommendation and ranking systems. They think they are interchangeable, but they are not actually. The ranking system depends on the queries based on the search of users who know what they are searching for, while the recommendation system does not take inputs from users and aims to show the things the users might not have found [1].

There is a massive growth in the use of recommendation systems these days. Commercial applications have the most use of recommendation systems. They are majorly used to recommend items such as products from e-commerce sites and are also used to recommend movies. It is one of the ways to increase sales without putting extra effort into marketing. Once an automated recommendation system is set up, additional sales automatically recur without effort. Recommendation systems help to make sales easier by recommending good options before customers even start looking for products. By getting customers to spend more time on their website, you can increase their understanding of your brand and interface, making them more likely to purchase in the future. Some big companies that use so much of recommendation systems are Amazon, Netflix, Spotify, LinkedIn etc. Fig. 1 shows the brief image of what a recommendation system is. [2].

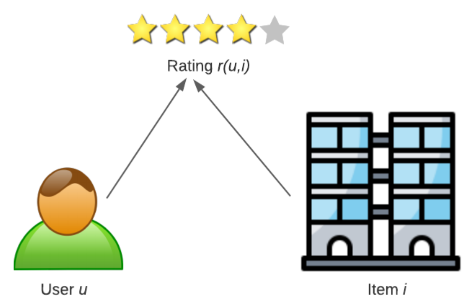

Fig. 1: Recommendation System

There are three types of recommendation systems. These are content-based filtering, collaborative filtering and hybrid systems. Content-based filtering methods rely on the characterization of the item (not the user) and the practicality

979-8-3503-9874-8/23/$31.00 ©2023 IEEE

profile of the user. This model works best with known data about the item (main cast, release year, movie genre, etc.) and questions about how the user has interacted with the recommendation system in the past but without the user's personal information. Content-based recommendation system focuses mainly on item content and user interest. These systems recommend the products based on the user's interest. The content-based recommendation is majorly a user-specific learning problem to quantify user utility, such as ratings, likes and dislikes based on the item characteristics. Moreover, the utility u(c,i) of item i to user c is estimated based on the utility u(c,i) assigned to other items previously viewed by the same user c [3]. The collaborative filtering system is based on user-user interaction, item-item interaction and user-item interaction, making the recommendations personalized. Collaborative filtering is used where users' data, such as age, gender, occupation, etc., are known. However, the missing data for items are challenging to perform the feature extraction for the items of interest. In contrast to content-based approaches, Collaborative recommendation systems attempt to predict a user's utility to an item based on other users' past utility to the item [4]. Hybrid systems overcome the problem of cold start, where recommendation systems need the minimum number of inputs from the user. Hybrid systems make sense to consider a combination of these two methods. Hybrid systems implement both collaborative methods content-based separately and combine their predictions. A hybrid system integrates content-based capabilities into collaborative ways of working. This approach is called the model ensemble approach. One way to do this is to use user profiles to measure the similarity between the two users and use that similarity as a weight in the aggregation step of the collaborative approach [5].

Traditional recommendation systems are non-personalised recommendation systems. They are popularity-based recommendation systems that recommend the most popular products to the users, for example, top-10 products, top-selling food, and the most frequently watched movies. Non-personalised recommendation systems are not that feasible when compared to personalised recommendation systems. A personalised recommendation system analyses users' data, purchasing, rating, reviews and relationships with other users in more detail. In this way, every user will be able to get recommendations that are customised to them. We have focused on a personalised recommendation system in this proposed work.

In this proposed work, we have focused on the user-controlled aspects. We are recommending particular items. We also recommend the most important aspects of products that the user controls. In addition, we recommend specific actions to the facility (item) manager to personalise the user's experience consuming the item. We can only take some aspects from the user's knowledge to improve the experience with the particular item. Such as, in a movie, aspects such as the movie's plot and cast cannot be in the user's control. The aspects of our model are extracted from the reviews.

In addition to predicting the item's rating based on the item's aspects, we would also rank the item using a specific ranking algorithm. The top n items will be shown based on the algorithm, and precision@n, recall@n and F1@n scores will be calculated for the same to check the efficiency of the proposed model.

II. Motivation

The significant growth and demand for online services have increased the number of applications from the offline to the online world. Companies such as e-commerce sites, banking, service booking etc., are a few examples of this. The user usually searches for information on a limited number of resources. The information which is searched could be found by checking at most a few items. When we search the web for information results, we may get millions of results. This overload of information occurs due to the unstructured texts, and users should ask for key-word based searches. In this case, it would be more suitable to have search engines that can understand the semantics underlying free text contents to identify the web resources that provide the near-perfect answers to particular questions.

Online reviews are helpful information for users to decide among their options. Users express their opinions and emotions through reviews for various items and aspects, i.e., characteristics, features, attributes or components. Users have to read the reviews carefully to make the best decisions.

The recommendation system helps them provide personalised suggestions to assist the users in making decisions. They use ratings that users give to an item. In addition to ratings, recommendation systems could also analyse and use the opinions that are being expressed in reviews given by the users. Overall ratings do not show the details of the items, so it is better to exploit reviews of items for user preferences and item aspects and get informed recommendations.

We have observed that in the existing recommendation system, they have predicted only the overall ratings. We have tried improvising the model to get the item's information and give the rating. We also thought to give the rankings, which will help the users quickly get the product of their interest without searching much. Hence, we have implemented a ranking feature missing from the existing recommendation system.

We have developed the proposed model for aspect-based recommendations to give users personalised recommendations by taking on particular item aspects.

III. Literature Survey

In today's era, consumers do not want to think twice about switching to the second option to find the products or services that suit them the most. Consumers often find it challenging to identify the one item that he/she would like, and that is precisely where the recommendation systems come into the role. History of the user-item interactions, such as viewing history, ratings or purchase logs, etc., are used to model user preferences and item features. These features are widely used in recommendation systems, and the technique is known as collaborative filtering (CF). One of the significant limitations of the collaborative filtering technique is the cold-start problem which is about the model's inability to give better recommendations to users with few ratings.

Many recommendation systems rely on the information (dataset) provided by e-commerce or some review website like Amazon or yelp. This information is used as reviews by users interacting with the items. More than just staying at the surface-level word representation is needed to model the rich vocabulary of the reviews. It is imperative to move beyond it.

Let us have an example where the following two sentences contain the word 'fast': (1)" She runs very fast", and (2)" All that happened very fast, we did not even get a chance to put our point". It is clear that 'fast' has positive sentiment in the first sentence, but the same word, 'fast', has negative sentiment in the second sentence. It makes it clear that the sentiment polarity of a word depends on the sentence in which it is used. So, the model needs a flexible word representation scheme.

Currently, what most of the recommendation systems do is exploit/read the user's reviews and based on that, they recommend items to that user. These recommendation systems need to include identifying relevant aspects of a particular item. Crowd-based natural language explanation based on user reviews is a more intellectual attempt to fill the void. The approach here is different since natural language processing is automated instead of the scenario where workers manually annotated the sentences to be included in the explanations [6].

Context-Aware Recommender System (CARS) is also related to the proposed model. The aspects we considered are not only limited to contextual variables. A significant amount of work has been done in CARS, including the task concerning user reviews. Most of this work has been used to develop new ideas that extract contextual information from the reviews the user wrote and use that information to give a predicted rating. Classification and text mining techniques were also used for identifying the sentences in the contextual information-rich reviews. The authors used their idea for an application related to the hotel having the objective of the trip contextual variable. Other areas where it is used are for extracting contextual variables like a companion, occasion, location, time, etc., in applications related to restaurants. All this is based on NLP techniques. In one way or another, using extracted contextual variables improves the predicted ratings of items [7].

It is essential to understand that a particular aspect vital for some users may only be considered for some. Which aspect to emphasise more completely varies from user to user [8]. For example, a user may prefer a particular tourist place for its variety of food or unique taste. In contrast, another user prefers the same tourist place for its natural scenes, the activities there, etc. Similarly, as can be seen, when going to watch a movie, one wants a good storyline for a horror movie, but when it comes to an action movie, then its cast is the only thing that matters. So, it is clear that the significance of each aspect depends on the user-item pair. There is no point in analysing this dynamic behaviour to get an answer of why a particular user emphasises more on some aspect rather than the other. The authors left it to the model to take care of these dynamic and fine-grained user-item interactions, which will later be used for getting predictions.

When it comes to transparency, then most recommendation system fails. Usage of textual explanations is one way of more transparency in recommendation systems. In the datasets we have, when a considerable amount of content information is there, then to explain a recommendation, item aspects may be aligned with the user's preferences. Critique recommendations are also one of the processes which should be considered. This critique can be done while we point out the aspects of recommended items, which can be done with the help of contextual information and the data we discussed earlier in the transparency concern. For item-based CF, the static variant used, e.g. by Amazon ("Customers who bought this item also bought. . . ") is in fashion these days. For model-based methods, it is still tricky (work is still being done in this area) to improve transparency through explanations [9].

Deep learning techniques were always in talks but before the surge in utilising them for recommendations, extracting/training the various aspects from the user's reviews got more attention from the researchers. At the aspect level, every minute detail in user-item interaction is essential, and that is where Aspect-based Neural Recommender (ANR), a neural recommendation system comes into play. First, the problem statement was defined, the notations used were specified, and then the model and its flow were presented. After that, the attention-based module was considered, i.e., its description for training the user-item interaction. Then for different weightage of aspects for different users, a co-attention-based module was shown. Then how this dynamic behaviour determines/predicts the overall rating was shown. After all this, the Aspect-based Neural Recommender model was optimised [10].

In the simplified model, the performance could have been better. The main reason was that more interaction between different aspect levels needed to be. This reason leads to the model's inability to perform as expected on both datasets. Additionally, to increase the overall performance, accepting the slight variations in the representations of the word through a layer (aspect-specific projection) can be considered. Finally, the dynamic behaviour of user-item interactions is essential, as suggested by the results of the model variants.

IV. Methodology

The flow of the proposed model starts from the training set that is input into a recommendation algorithm that gives a model that will be further used for predictions. For evaluation, the model is fed with the test set, and predictions are made. Predictions are then fed to the evaluation algorithm to get the final results.

We have built our model on various aspects of an item. For example, let us consider the example of buying a house. Aspects (A) can be the location of the house, what is the size, the rent of the house, car parking, extra services etc. Aspects are things that can be derived from that object or considered.

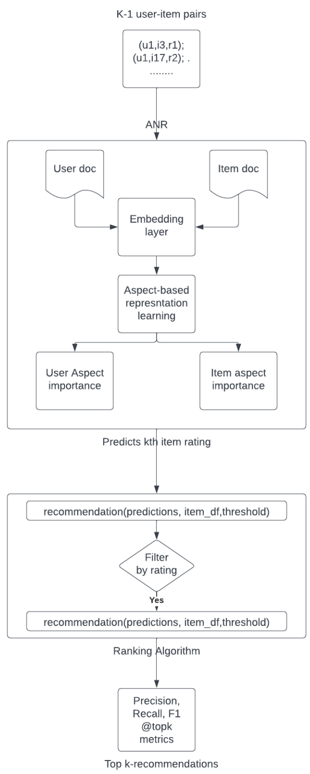

Fig. 2: Overview of proposed model

Now let us have an idea of the overall architecture of the model. We have user document representation *Mu*. We have to find a set of aspect-level user representations *Pu* = *pu, a* — *a A* w.r.t., a set of K domain-dependent aspects, A. User opinion towards an item is described in the review *du, i*. The properties of a particular item w.r.t a are better described by the item document (*Di*). Some users may have reviewed it, and based on that, and the properties revolve around them. Du is the user document which contains the user's opinion towards the items u has bought or interacted with earlier. From the user-item interaction document, we have to find the aspects from all the data we have. We will derive a set of aspects "A" from that. We have proposed aspect based learning method. The main focus here is to have a rich user-item interaction. The relation between user-item importance is dynamic.

The model performance can get affected by various parameters. Here, the qualitative analysis of the learned aspects will show how the proposed model works.

Parameter sensitivity: It plays a significant role in the accuracy of our model. A varying number of aspects, some factors etc., comes under this. In different data sets, the number of optimal aspects varies, and the characteristics of the review content depend on a given data set. While measuring the proposed model's performance, we concluded experiments that by keeping the number of aspects around 4-6, we get a reasonably good performance. Increasing or decreasing the total number of aspects will have some effect on the granularity of every single aspect, that is, a large number of precise aspects v/s a small number of aspects on a broader level. However, the model's overall performance gets very little affected by increasing or decreasing the total number of aspects, but this happens only within a reasonable minimum to maximum change. H1 (On the aspect level, to represent items and users, latent factors have to be considered, and their number is H1) and H2 (Based on the affinity matrix, estimations regarding the users and aspect's importance are made, and hidden layers are used for this and its size is H2) on getting varied also has an impact on the performance of the model. We found that keeping the value of H1 higher than 15 does not improve model performance because the model only needs a small number of latent factors to cypher aspect-level representations of user items. However, having insufficient latent factors harms the performance of the model.

User-item representations are then derived by putting user-item representations into a hidden layer. When deriving an overall rating, a particular user and an item have an interaction between them. The pre-training stage provides a better starting point for establishing the importance of user and item aspects. To further improve performance, we allowed variation in word representation through an aspect-specific projection layer. All these results conclude that for each user-item pair, there is a need to adjust the importance of the user-item aspect dynamically. The overview of our model is shown using a flowchart in Fig. 2.

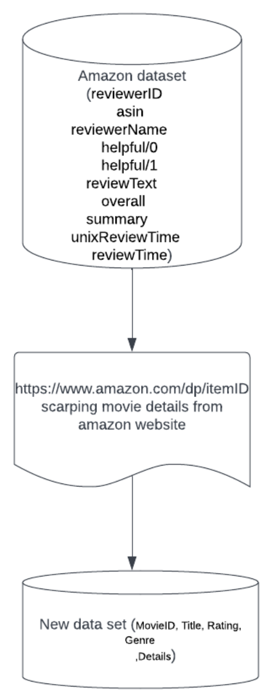

Fig. 3: Web Scraping Flow

Now, from the model, we got the predicted rating and predicted Movie-ID that will be used to get the top best recommendations of movies. Now the dataset from web scraping will be used. The flow of how weeb-scrapping is working is shown using fig.3 for the refrence while web-scapping dataset is explained in later section. The Amazon dataset we used has a size of around 37,000 entries, out of which we used around 20,000 to train our data. The dataset we got from web scraping is around 500, of which we used nearly 70 entries for testing purposes. Data Frame is constructed using the Python Pandas library. Using the predicted movie ID, we will first extract that movie's features from the dataset and then filter movies that take place according to their genre and ratings. If the genre of predicted movies and the movies in our dataset match, that movie is added to our new shortlist movies list. After this, filtering based on rating took place. If the movie rating in the shortlist is greater than or equal to the predicted rating, then that movie is considered. Finally, all the movies we have on the list now will be sorted in decreasing order of rating. This was the ranking algorithm, and the output will be the file that contains Movie-ID and ratings after all the filtering, which will be further used for precision-recall analysis.

After getting the predicted item ids, user ids and ratings, we calculated the Precision for top k items (precision@k) and the recall for top k items(recall@k). Recall and Precision are the key elements to check for retrieving the information. The positive class was most important compared to the negative. When searching for something on the web, the model does not care about something irrelevant and not loaded (this is the true negative case). Therefore, only TP, FP, and FN are used in Precision and Recall. What percentage of all positive predictions are positive is nothing but accuracy. The accuracy value lies between 0 and 1. On the other hand, the percentage of the total number of positives is predicted as positive. It is the same as TPR (true positive rate).

Many evaluation metrics are available for recommendation systems, and each has advantages and disadvantages. For example, RMSE can be calculated by comparing the predicted rating with the actual rating for each user-item pair with a known label. Precision and recall are binary metrics used to evaluate models with binary output. So we converted our numerical problem (ratings usually from 1 to 5) into a binary problem (relevant and irrelevant items). When translating, we will assume that any actual rating above 3.5 corresponds to a relevant item and any actual rating below 3.5 is irrelevant. A relevant item for a particular user-item pair means that the item is a good recommendation for that user. (3.5 is just the threshold I chose. There are several ways to set this threshold, such as taking into account the user's rating history).

V. Experimental Results

A. Datasets

We have evaluated our model using the publicly available datasets from Amazon of movies. The other dataset is obtained from the Amazon website by web scraping. The description of both the datasets is shown using two tables i.e. Table I and Table II. This section will describe the datasets used, their contents, and how we got them.

RS datasets are typically spare in applications that operate within an immediate time frame. This spareness leads to another issue. This issue may be summarised by the inability of the system to give a good and effective recommendation to the users. There are only limited data for ratings, and most of the time, items or users are posed only with biased terms. Using text reviews and extracting information from user text reviews, we have tried to overcome this issue with our proposed method since the user/item side information is there in the review [11].

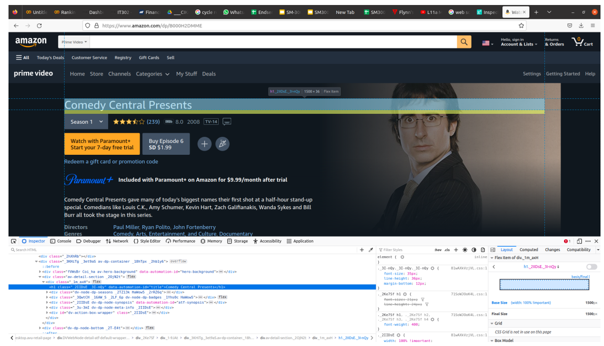

Fig. 4: Amazon WebPage

1) Amazon Movie DataSet: We obtained this review dataset online, containing 20000 entries from May 1996 to July 2014. This dataset contains Reviewer-ID, Item-ID, the name of the reviewer, helpfulness of rating, the review is given, time of review and a little summary about the review. Dataset is cleaned, and all the duplicate entries are removed from them. This is also shown using Table I. A sample movie review data is mentioned [12].

where,

reviewer ID is the ID of the reviewer.

asin is the ID of the product.

reviewerName is the name of the reviewer

helpful is the helpfulness rating of the review, e.g., 2/3

reviewText is the text of the review

overall is the rating of the product

summary is the summary of the review

unixReviewTime is the time of the review (unix time)

reviewTime is the time of the review (raw)

2) Web scraping Movie Data: Web scraping is a method by which we can extract data from websites. There is a library named BeautifulSoup in Python. This library is used in web scraping to retrieve data from the website. We used it to scrape movie information and saved the details in a CSV file. We have obtained Movie ID from Amazon Movie Dataset(previous dataset) and used it as input here to get details of those movies from the Amazon website. We feed the URL from Movie-ID to our soup object, extracting the required information from the given URL link based on the movie id, making a new CSV file, and saving it to a new CSV file. The Amazon movie dataset contains reviews, so each movie has multiple reviews. Details of all the unique Movie-ID are obtained, and the unique movies were 994 in the count. We have scrapped the movie

TABLE I: Amazon dataset

reviewerID	asin	reviewerName	h/0	h/1	reviewText	overall	summary	unixime	reviewTime
A11N155CW1UV02	B000H00VBQ	AdrianaM	0	0	I had	2	A little	1399075200	05 3, 2014
A3BC8O2KCL29V2	B000H00VBQ	Carol T	0	0	I highly	5	Excellent	1346630400	09 3, 2012
A60D5HQFOTSOM	B000H00VBQ	Daniel Cooper	0	1	This	1	Way	1381881600	10 16, 2013
A1RJPIGRSNX4PW	B000H00VBQ	J. Kaplan "JJ"	0	0	Mysteries	4	Robson	1383091200	10 30, 2013
A16XRPF40679KG	B000H00VBQ	Michael Dobey	1	1	This	5	Robson	1234310400	02 11, 2009
A1POFVVXUZR3IQ	B000H00VBQ	Z Hayes	12	12	I	5	I purchased	1318291200	10 11, 2011
A1PG2VV4W1WRPL	B000H0X79O	Jimmy C. Saunders	0	0	It	3	It	1381795200	10 15, 2013
ATASGS8HZHGIB	B000H0X79O	JohnnyC	0	0	There	3	A reaso	1388275200	12 29, 2013
A3RXD7Z44T9DHW	B000H0X79O	Kansas	0	0	This	5	kansas001	1393372800	02 26, 2014
AUX8EUBNTHIIU	B000H0X79O	Louis V. Borsellino	0	0	Not bad.	3	Entertaining C	1396396800	04 2, 2014
AXM3GQLD0CHIL	B000H0X79O	Ray	0	0	Funny	4	Worth	1391731200	02 7, 2014
A398QSASJOIKA6	B000H29TXU	Amazon Customer	0	0	I love	4	comedy	1391644800	02 6, 2014
A2U61O0KWJH3MM	B000H29TXU	Cathy P.	0	1	comedy	3	ok	1378339200	09 5, 2013
A2LSZFEFTDRDIJ	B000H29TXU	debra marrero	0	0	If this had to	3	not	1371168000	06 14, 2013

TABLE II: Web scraping dataset

MovieID	Title	Rating	Genre	Details
B000I8FOZA	Extreme Engineering	3.9	Kids, Documentary, Special Interest	Extreme Engineering........
B0083IJGBU	Pride and Prejudice (1995)	4.9	Drama, Romance	Witty Elizabeth........
B001QDT85I	Combat Zone	4.4	Historical, Documentary, Military and War	Each episode......
B004VJQ7F8	Workaholics	4.6	Comedy	Join Adam......
B009PI7EIE	The League	4.8	Comedy, Sports	To be......
B007UXTC2M	Without Motive	3.7	Suspense	DC Jack Mowbray.....
B000H00VBQ	Wire In The Blood	4.4	Suspense, Drama	A clinical.....
B000I5Q0ZG	Doctor Who: The 50th Anniversary Collection	4.5	Suspense, Drama	The first.......
B004WYXHHI	Mob Wives	4.7	Unscripted	Welcome to......
B009VSK00W	Don't Trust The B—- In Apartment 23	4.8	Comedy	The hilarious.....
B000OFME82	The Sarah Silverman Program	4.5	Comedy	The Sarah Silverman......
B007CUQ82A	Monarchy with David Starkey	4.5	Historical, Documentary	The English......
B007YP4ZLY	Flipped Off	3	Unscripted, Drama	Throwing down......
B006S0007E	Thor & Loki: Blood Brothers	4.5	Suspense, Animation, Drama, Anime, Fantasy	The mighty......
B006H0602Y	Roadie	3.7	Comedy, Drama, Music Videos and Concerts	Canned from.....

title, average rating, genre, and some descriptions of the movie, which can be shown in the fig. 4. Moreover, all these data are taken from the official Amazon website. This dataset will be mainly used during the ranking algorithm and will help to get aspects of the movie. The dataset we got from web Scrapping is shown using Table II.

B. Results

For the pre-training part, we tried various combinations by varying loss function(-loss-function), no. of epochs (-e), learning rate(-lr), optimizer(-opt), dropout rate(-dropout-rate) and the number of aspects(-K). The loss function used was MSE in all the cases. We recorded different values of training loss, MAE value and MSE value. The default value of batch size is set to 128 in our model, while the number of aspects was set to 5 by default. We have shown the performance metrices is terms of values of precision , recall and F1 score for the top k values. This is shown using Table III.

TABLE III: Performance metrices for top k values

k	Precision@k	Recall@k	F1@k
5	0.44205	0.17901	0.25172
10	0.44204	0.35802	0.38973
20	0.32298	0.44205	0.36744

The best case was recorded when we ran the query for the number of aspects equal to 5, batch size equal to 64, and learning rate equal to 0.002. We recorded the values for 30 and 50 epochs. When 30 epochs were run successfully, the training loss recorded was 0.60473 and [DEV] MSE: 0.97727, MAE: 0.73039 and [Test] MSE: 1.04483, MAE: 0.74688 recorded. Running the experiments with the same combination for 50 epochs the training loss recorded was 0.56988 and [DEV] MSE: 1.00384, MAE: 0.73799 and [Test] MSE: 1.08390, MAE: 0.75691 is recorded.

VI. Conclusion and Future Scope

We have extended and proposed work based on the neural attention and co-attention mechanism. We have proposed an "Aspect-based neural recommender (ANR) using adaptive prediction," including an aspect-aware representation learning component and an aspect importance estimator. After carrying out the experiments, we found a significant improvement over existing state-of-the-art recommendation systems that the ANR achieves statistically. We have done web scraping for datasets and got them from Amazon reviews. Apart from that, we have significantly increased the precision for top k (precision@k) and decreased the recall for top k (recall@k). We have also used the ranking algorithms to get the predictions, i.e., a list of item-id and item-rating.

To enhance our model further, instead of just getting predicted ratings, we also used the function (top-n-recommendation) to get top-n recommended items for a particular user. We can extend ANR into a domain-independent framework to handle multiple categories simultaneously. This future direction can be implemented by incorporating either transfer learning or multitasking. Apart from this, we attempted to add item id as a predicted feature that we thought would help to give better recommendations, but it could not.

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*. Springer, 2011, pp. 1–35.

[2] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics reports*, vol. 519, no. 1, pp. 1–49, 2012.

[3] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop*, vol. 30, 2000, pp. 47–56.

[4] J. Basilico and T. Hofmann, "Unifying collaborative and content-based filtering," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 9.

[5] Y. Li, L. Lu, and L. Xuefeng, "A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in e-commerce," *Expert systems with applications*, vol. 28, no. 1, pp. 67–77, 2005.

[6] C. Musto, G. Rossiello, M. de Gemmis, P. Lops, and G. Semeraro, "Combining text summarization and aspect-based sentiment analysis of users' reviews to justify recommendations," in *Proceedings of the 13th ACM conference on recommender systems*, 2019, pp. 383–387.

[7] K. Bauman, B. Liu, and A. Tuzhilin, "Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 717–725.

[8] V. V. Kumar, K. Raghunath, V. Muthukumaran, R. B. Joseph, I. Beschi, and A. K. Uday, "Aspect based sentiment analysis and smart classification in uncertain feedback pool," *International Journal of System Assurance Engineering and Management*, vol. 13, no. 1, pp. 252–262, 2022.

[9] T. Donkers, T. Kleemann, and J. Ziegler, "Explaining recommendations by means of aspect-based transparent memories," in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 2020, pp. 166–176.

[10] J. Y. Chin, K. Zhao, S. Joty, and G. Cong, "Anr: Aspect-based neural recommender," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ser. CIKM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 147–156. [Online]. Available: https://doi.org/10.1145/3269206.3271810

[11] A. Da'u, N. Salim, I. Rabiu, and A. Osman, "Weighted aspect-based opinion mining using deep learning for recommender system," *Expert Systems with Applications*, vol. 140, p. 112871, 2020.

[12] S. A. Rahin, T. Hasib, and M. Hassan, "Aspect-based sentiment analysis using semeval and amazon datasets," in *2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*. IEEE, 2022, pp. 85–90.