Personal recommender system via convolutional auto encoder with conditioning augmentation

Recommender system and representation learning with convolutional autoencoder

Sunjun Kim

Department of Industrial engineering

Ulsan National Institue of Science and Technology

ULSAN NATIONAL INSTITUTE OF
SCIENCE AND TECHNOLOGY

Personal recommender system via convolutional auto encoder with conditioning augmentation

Sunjun Kim

A thesis presented to
Ulsan National Institute of Science and Technology
for the degree of master

Department of Industrial engineering

12.21.2022 of submission

Approved by

임 성 훈

Advisor

Sunghoon Lim

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

Personal recommender system via convolutional auto encoder with conditioning augmentation

Sunjun Kim

This certifies that the thesis of Sunjun Kim is approved.

12.21.2022 of submission

Signature of Advisor: Sunghoon Lim

임 성 훈

Signature of Committee: Sangjin Kweon

Signature of Committee: Gisoo Kim

Abstract

Recently, volume of various types of information, including reviews, images, and videos containing sound, also known as unstructured data, have been increased in an explosive manner [1]. Even though, unstructured data is difficult to utilize without proper preprocessing, many applications adopted it as a source of information to extract value from it, such as recommender systems, natural language processing and computer visions [2]. The recent prosperity of deep learning techniques has accelerated the progress in this field by making the preprocessing parts and feature extraction parts simple and easy [3]. Moreover, the emergence of generative adversarial network has led to improved general performance of unsupervised learning models which makes many applications to make use of diverse forms of data

Along with this context, concerns of this article focused on (i) developing a recommender system based on modified autoencoder which is a typical deep learning technique applied to this research field that presents exceptional performance in the feature extraction process and (ii) applying data augmentation to this field which is frequently used to deal with data scarcity or limitation problem which is one of the main challenges of recommender systems [3].(iii) Lastly, the proposed model can be applied to both tasks, collaborative filtering and contents-based filtering proved to present compliant performance

Modified convolutional autoencoder-based recommender system learns features of samples that represented by reviews of users or user-item rating matrices. The proposed model takes vanilla autoencoder as a base structure conbined with convolutional layer to extract feature that takes encoded vector as input which is represented by preprocessed reviews or ratings. Afterward, the conditioning augmentation process, which is an augmentation technique for embedded vectors, is applied that goes through a decoder that produces final predictions based on the encoded input vector from the previous encoder

The contribution of the paper can be summarized as three points. (i)Conditioning augmentation which is data augmentation technique utilizing encoded vector to deal with data scarcity problem that is mainly concerned in recommender system field. (ii)Proposed

model can take both types of inputs which are contents-based encoded vector or rating-based encoded vector. Contents-based vector can represent features of item such as review, quality, and various types of categorical feature to corresponding product. Rating-based vector indicates evaluation of consumer based on numeric value. (iii)Lastly, performance of proposed model is compliant compared to state of the arts of several bench mark open dataset.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The exponential growth of information and the number of users of the internet and various types of crowdsourcing platform, not only made tremendous opportunities for business providers, but also have presented different kinds of challenges by information overload [4]. Major companies such as Google, Amazaon, and Facebook have treated these challenges by utilizing information retrieval systems by collecting and managing information from billions of active users. However, without prioritization and customization of this overflowing information, creating proper values for consumers was not implementable. This phenomenon has naturally led to an increase in demand for recommender systems, which are useful algorithms to filter and provide proper information to proper consumers based on their preference and priority [5].

The purpose of recommender systems is to provide relevant item lists or services to users based on their inclinations. The systems are not only beneficial to providers but also to consumers by reducing the cost of searching, selecting and other various types of transaction costs [6]. For example, in e-commerce situations such as Amazon, it has been proved that the recommender systems improve the quality and quantity of transaction processes related to the profit margin of the business and consumers' experiences [1].

In a similar vein that most machines and algorithms have their own limitations or shortcomings, several types of limitations and challenges exist in recommender systems, including cold start problem, scalability, interpretability, and data requirements. In addition, considering the size and the complexity of information that the system needs to handle, the difficulty of solving problems becomes harder [7]. This paper mainly aims to deal with the data scarcity problem, which is about the robustness of the proposed model. Excluding major platforms such as Youtube, Netflix, and well designed open datasets for recommender system, quality and quantity of user-item information are limited. Therefore, we believe that solving the data requirements problem will propose proper value and contribution to the application of the recommender systems.

1.1.1 Ubiquity of deep neural network and recommender system

As various types of deep neural networks are applied to providing online services such as over-the-top (OTT) service, recommender systems account for a huge part of online business companies [8]. More than 50 percent of videos are watched via the recommender systems and approximately 70 percent of movies and video contents are visited by consumers through the recommender system. In addition, the demo graphical range of users on Youtube and Netflix has increased a lot, meaning more users are exposed to recommendation services more than ever.

Subsequently, research related to deep learning based recommender systems has increased both quantitatively and qualitatively. For example, published wide and deep network that proposed a solution to the overfitting and general suggestion problem of recommender systems by combining wide components and deep components [9]. On Netflix, which is the most famous and the largest OTT platform, most of the listing algorithms shown on the service are based on recommender system algorithms such as personalized video ranker(PVR) and Top-N video ranker. Moreover, the number of papers submitted to RecSys, which is the conference hosted by ACM related to the recommender system, has increased to 66 percent [4].

Modern recommender systems consists of four architectures as shown in Figure 1, which are user-item database, preprocessing, feature extraction, and recommendation service. As mentioned above, the prosperity of deep learning techniques has accelerated the research field of recommender systems, especially on feature extraction part [10]. For example, deep learning techniques such as autoencoder, generative adversarial network, and multi layer perceptron have shown peculiar performance on learning representation from user item information that is traditionally performed by matrix factorization. Moreover, as convolutional neural network (CNN) presents exclusive performance in dealing with unstructured data which is known as unsupervised learning like images and texts, the preprocessing part also benefited from deep learning to some extent.

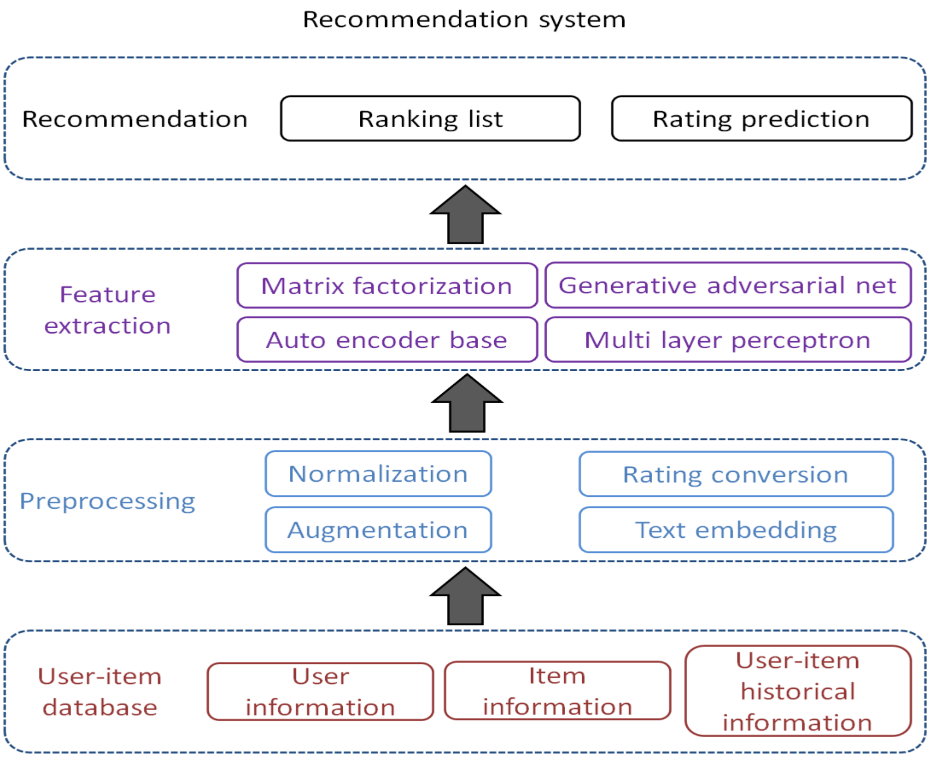

Figure 1: Modern recommender system

1.2 Unstructured data

As mentioned above, data pools of recommender systems are usually huge and exist in diverse forms such as images, texts, sounds, and videos that are combined forms of previous ones [11]. This section focuses on various types of unstructured data and corresponding preprocessing to generate valuable input sources for recommender systems [12].

1.2.1 Image data

Image data is another fine example of unstructured data that can be obtained through various types of channels such as croud-sourcing platforms, social network services and every kind of sensor that can take pictures. The fundamental unit of image data is pixel, which is represented as a numerical value that has a range between 0   255 in most cases [13].

As computer vision is one of the largest fields in deep learning research, image data also has the widest range of applications and specific research areas. Object detection, segmentation, and pose estimation are typical examples of computer vision tasks that use image data as input resources for training models [14]. Thereafter, many applications and techniques have been developed related to dealing with images.

1.2.2 Text data

Text data indicating documents, reviews, or even scripts in movies are typical examples of unstructured data. In most cases, text data consists of a corpus that contains various types of meta data, linguistic features, and semantic features that can be used in many applications [15].

For example, natural language processing, which takes a large proportion in artificial intelligence research fields, uses text data as the main resource of information. Text classification, machine translation, and text summarizing are included in this category which are recently accessible easily in real life [16]. In many cases, corpus, document, and tokens are fundamental units of inputs in this task. The smallest unit is a token that can be a word from a newspaper or a sentence from an article. A group of tokens forms a document and a group of documents forms a corpus.

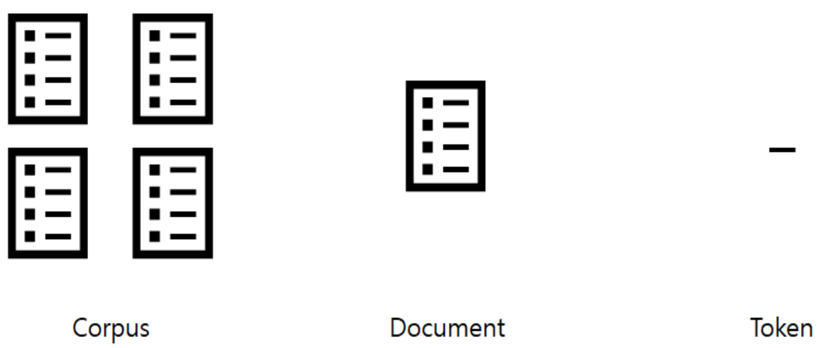

Figure 2: Heirarchy of unit in text data

1.2.3 Tabular data

Tabular data consists of columns and rows. Each column indicates a feature of corresponding data, and each row indicates a sample. Most rating-item matrices are tabular data which is frequently used for predicting ratings of users to corresponding item sets [17]. Traditional recommender systems based on matrix factorization or auto-encoder mostly use this type of data to predict the recommendation list using rating values.

User/Item	Item A	Item B	Item C	Item D	Item E	Item F
User1	2	3	4	4	3	1
User2	2	1	5	5	2	3
User3	3	2	1	2	1	3
User4	3	4	0	3	3	4
User5	2		3	3	2	2
User6	3	3	0	3	3	1
User7	5	2	1	1	3	5
User8	3	4	2	1	3	5
User9	1	3	2	2	1	4
User10	3	1	2		3	2
User11	2	5	3	2	3	4
User12	1	3	2	3	2	2

Figure 3: User-item matrix

1.2.4 Video

A video is a combined type of data that contains text, image, and sound in one form. YouTube content can be a good example of video data as it contains titles, subtitles which could be classified as text data, sequence of frames as image data, and sound as it plays the content. Moreover, video data provides other various types of information such as number of view, watch history of each user, number of likes and dislikes of a corresponding video. Recently, as the number of companies providing over-the-top media service, also known as OTT service, has increased, the pool of video data is becoming larger in a rapid manner and this can provide plenty of information to recommender systems while it requires a high degree of refining process [18].

1.2.5 Preprocessing

Text data contains various kinds of noise such as punctuation, emotions, and several types of abbreviations, which have prevailed recently as social network services have developed. To feed data into the model, data needs to be cleaned by tokenization, vectorization, normalization and etc [19].

Preprocessing of image data includes not only typical preprocessing of deep learning techniques such as normalization and vectorization, but also specific processes which are geometric transformation, augmentation, and so forth [19].

Numeric data such as rating-item matrix requires less preprocessing than other types of data such as image, text or categorical features [20]. Proper normalization and residing of the tensor are sufficient for it to be input of the model.

Video data requires the highest degree of preprocessing among motioned datasets, because it is a sequence of image frames. Not only does it require preprocessing of image data, but also the order of frames, shape of frames, and frame rates should be included in the process [21]. For example, in terms of the 3D pose estimation task, which is about predicting 3D key points axes of human joints, the number of frames to predict one key point should be considered and it takes an important role related to the performance of the model [22].

1.3 Recommendation system

1.3.1 Collaborative filtering

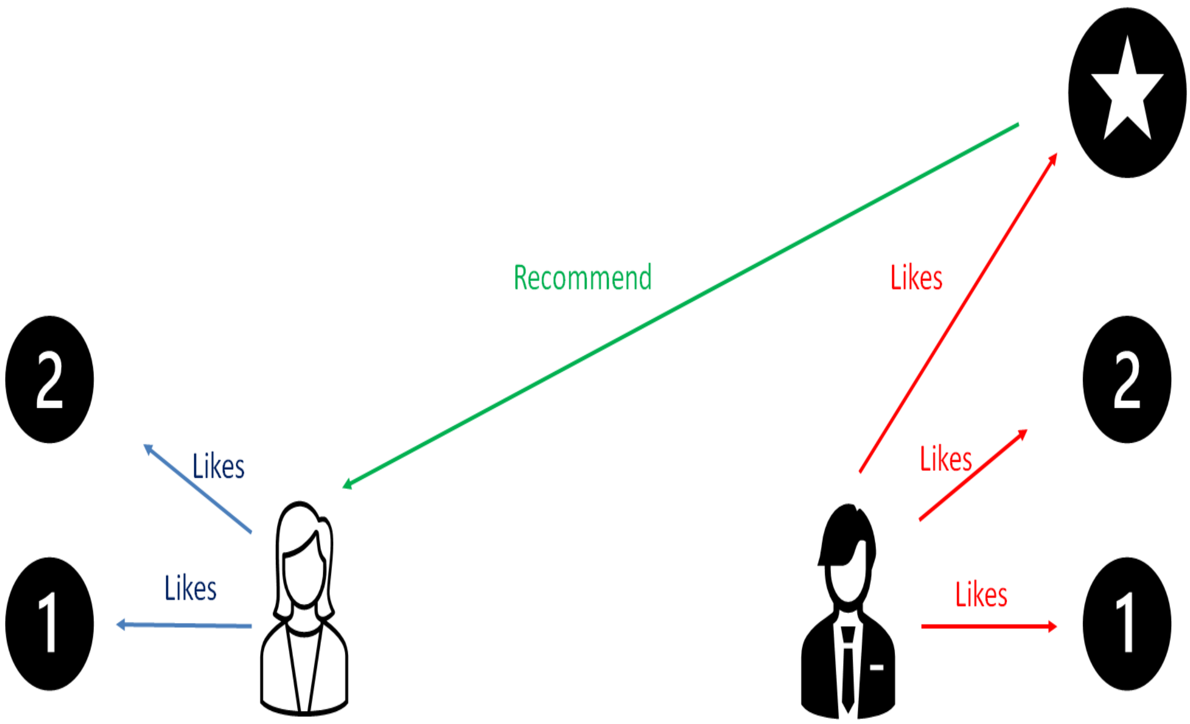

Figure 4: Collaborative filtering recommendation

Collaborative filtering is the most frequently used technique in recommender systems, which makes recommendation lists based on similarity between users. The main assumption of this technique is that similar evaluations of consumed products can represent similar preferences of users [23]. Mostly, the evaluation of users is measured by ratings of the items provided. As shown in Figure 5, this algorithm can recommend an item to user A based on other user B who has similar interests to user A. Another advantage of the algorithm is that feature engineering does not need to be implemented manually [24].

1.3.2 Content-based filtering

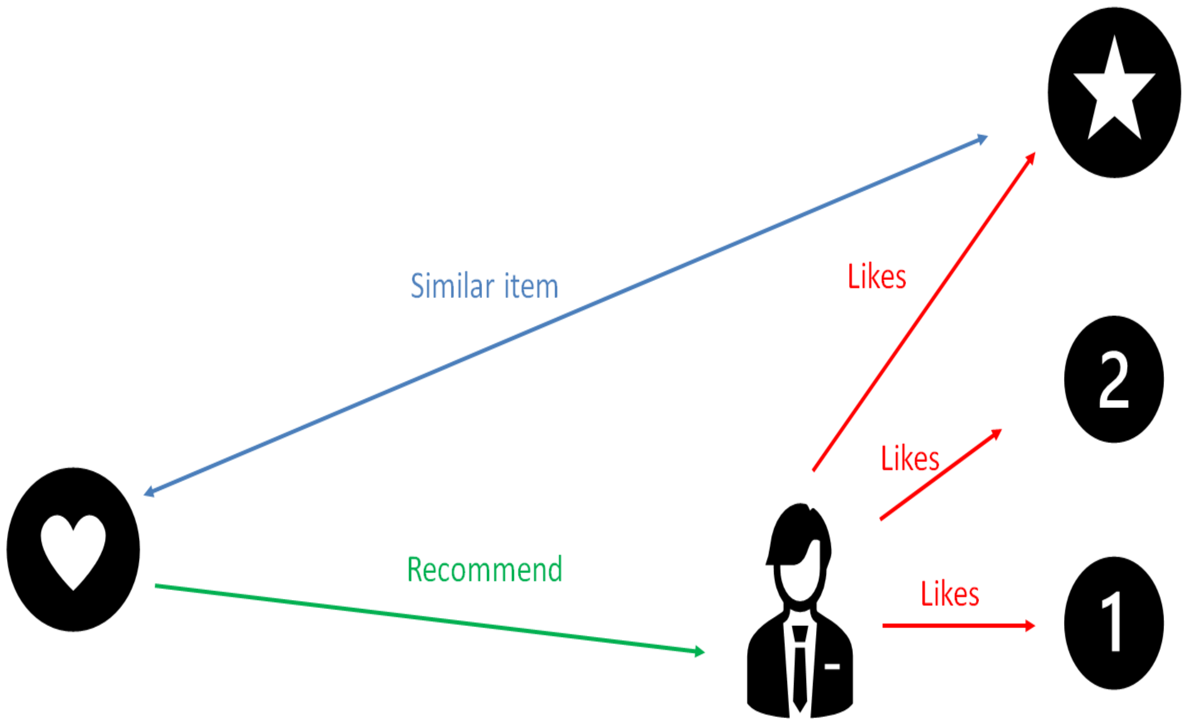

Figure 5: Content-based filtering recommendation

The main concept of content-based filtering technique is based on similarity between items. By analyzing and learning the features of items, this technique determines the recommendation list of a specific user's preferred items [25]. Moreover, content-based filtering is appropriate to capture specific preferences of users and is able to predict item lists that are not preferred by other users in usual [26]. However, the main limitation of this technique is that it requires a high degree of domain knowledge to learn representation of items. For example, to represent a car, service providers have to distinguish what features may or may not affect the preferences of users and features of items itself [27].

1.3.3 Main challenges in recommender system

- Data sparsity. Even though, size of the data set is large enough to train the recommender system, the dataset itself can be very sparse, because not all consumers can use all products provided by companies. This aspect can degrade the system's performance, which is why the recommender system requires proper robustness to data sparsity [6].

- Data requirement. The recommender systems play an important role in business, especially for online service providers and it comes under not only for major companies but also for minor companies that have a short-term business history [6]. In the case of minor companies, lack of proper tools for data management can induce low quality and quantity of recommender system source which is critical.

- Scalability. As recommendation techniques and crowd-sourcing platforms which are the main sources of big data prosper, size of the dataset keeps increasing in an exponential manner [28]. Traditional recommender system techniques such as matrix factorization can face scalability problems as computational costs get too high because of increased dataset size.

- Cold start problem. As mentioned in data management limitation, most startups have short business history, which means the companies do not have much information about users [29]. This aspect can also affect the performance of the recommender systems.

2 Related work

In this section, various types of deep learning techniques used in recommendation systems will be introduced with brief explanation, including multi layer perceptron, autoencoder, joint learning, and generative adversarial net. Moreover, category table that contains recent publications related to corresponding techniques of current trends will be provided.

2.1 Collaborative filtering with neural network: Netflix, personalized video rank

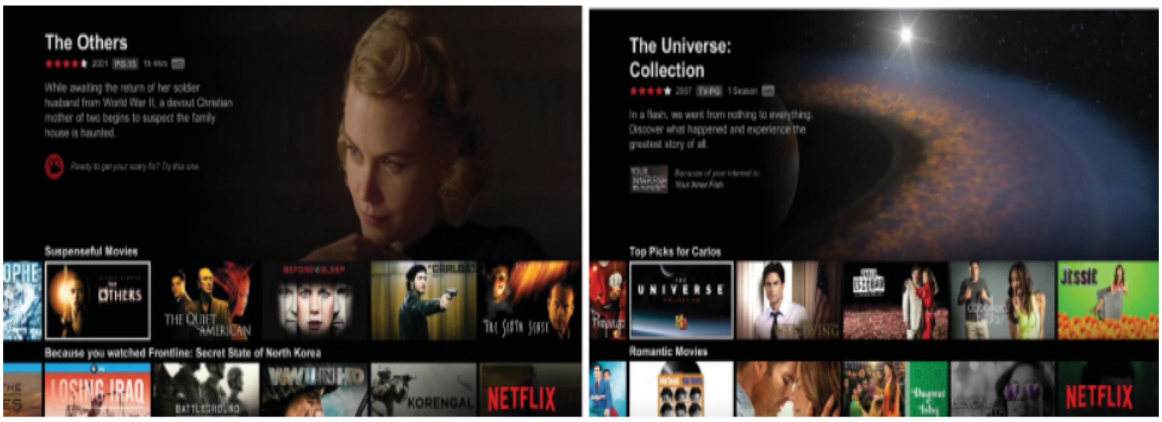

Figure 6: Recommendation example page of Netflix (Carlos A et al. 2015)

As mentioned above, Netflix is a major OTT service provider. Most of the algorithms contained in the recommending services of Netflix are based on recommendation system techniques with deep learning, including personalized video ranker (PVR) and top-N video ranker [30]. As the name of the algorithm suggests, PVR provides video aligned sets of items considering the personalized profile of each user. So, PVR can be described as a filtering system to provide a delicately customized recommended list [30]. On the other hand, the top-N ranker algorithm suggests several items that are picked mostly at that time, which means it considers the general preference of users at that time more than the PVR algorithm. These two algorithms are designed with multi layer perceptron that takes several watch, search vectors as input to predict ratings of items.

2.2 Contents based filtering with auto-encoder, Samsung

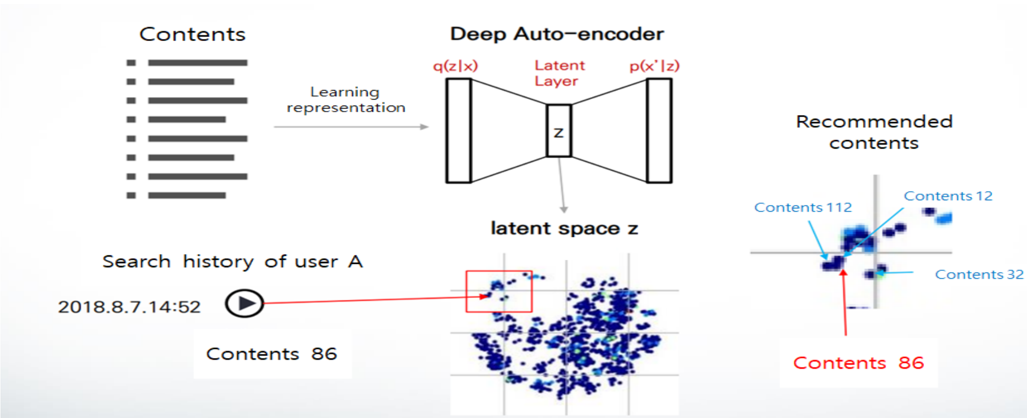

Figure 7: Video recommendation system of Samsung (Samsung techtonic workshop. 2018)

The popularity of contents creating businesses is growing rapidly in the modern virtual market. Generally, users on the internet make the decision to watch videos based on their preferences and relative needs [31]. A recommendation system is crucial to meet the needs of users on online platforms [32]. Samsung suggested contents based filtering with autoencoder learning representations from processed features of videos. Moreover, features of videos are extracted from diverse sources such as scripts from the video, title, and video frames, which require various types of preprocessing technique

As the autoencoder learns representation from inputs, it generates latent space of z which represents each item. In the next step, recommendation sets are extracted based on cosine similarity considering input features are composed of text data.

2.3 Joint learning of deep and wide component, Google

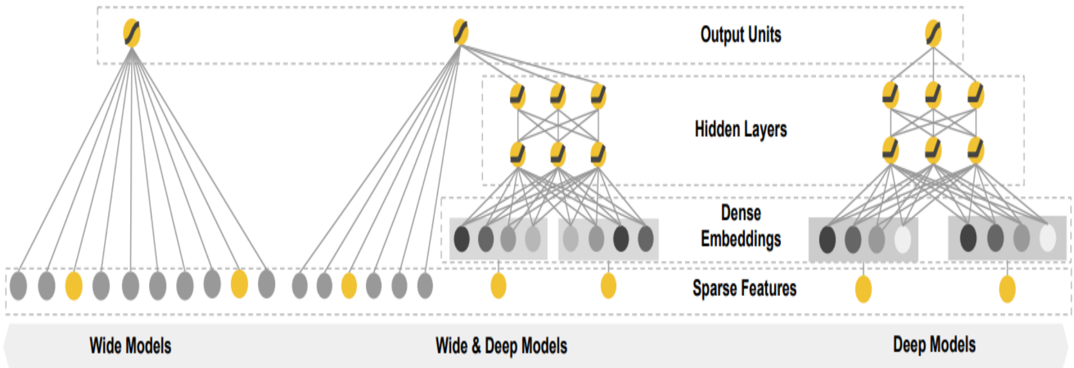

Figure 8: Wide and deep network of Google recommendation system (HT Cheng et al. 2016)

Before the trend of joint learning recommendation systems, sparse input data was used to predict large scales of ratings or recommendation lists with a combination of linear models and non-linear feature transformations [9]. In the case of generalization components, it requires more effort and cost for feature engineering, but the emerging of deep neural networks dramatically reduces the cost of feature engineering. However, deep neural networks can be too generalized with sparse inputs and low dimensions, which is also known as overfitting. To supplement this problem, memorization of feature interactions through wide components such as cross-product feature transformations is used. As a result, jointly trained models with deep and wide components can exploit both benefits of memorization and generalization [9].

In addition, the system was productionized and evaluated on a huge e-commerce platform like Google Play, and a commercial mobile app store with approximately a billion active customers that have made the system more robust and better.

2.4 Recommendation system with generaive adversarial net

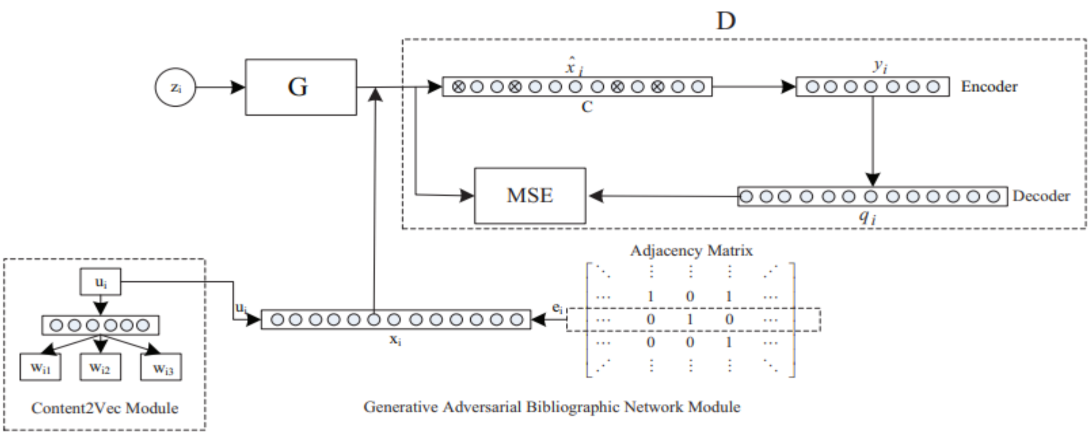

Figure 9: GAN for recommendation system (Xiaoyan Cai et al. 2018)

As generative adversarial network (GAN) emerged, representation learning by generative model has been recently utilized for many applications and the recommendation system is also one of them [33]. GAN is composed of two structure, generator and discriminator. A generator learns the distribution of corresponding input variables to replicate new generated data [1]. A discriminator works as a detector to determine whether input variables are generated or origin. By training these two networks simultaneously, the generator is able to synthesize new data that is almost similar to the origin one. Based on the model, GAN-based recommendation system can learn an effective feature representation space that preserves both contents information and network structure information that can be applied to personalized recommendation task.

3 Proposed model

3.1 Model description

This section presents not only the proposed model but also the overall modeling process to handle the data limitation problem. Moreover, the actual learning and prediction process of the model will be provided.

3.1.1 Application of autoencoder to collaborative filtering

In most cases, autoencoders usually replace matrix factorization of recommender systems. However, the main limitation of this task is that input data $x$is a sparse matrix, which means it contains many missing entries. Meanwhile, the output from the decoder is a dense matrix. The learning process of the autoencoder to learn representation from the user-item matrix is based on dense refeeding process and masked mean squared error loss function (MMSE) [34] [35]. The calculation of MMSE and the overall process of learning can be described as follows.

$$Loss\,function\,MMSE = \frac{m_i * (r_i - y_i)^2}{\sum_i^n m_i} \tag{1}$$

- r_i is an actual rating
- m_i is 0 when $r_i = 0$, or 1
- y_i is a predicted rating

Dense refeeding process

1. Dense output decoder $f(x)$ is calculated using 1 given sparse input data x (forward)
2. Updating weights based on the output from 1 (backward)
3. Predicting $f(f(x))$ while assuming $f(x)$ as an new sample
4. Updating weights based on the output from 3 (backward)

This dense refeeding process can be implemented more than one time for each training epoch.

3.1.2 Denoising autoencoder

The first approach to solving data limitation was taken by utilizing denoising autoencoder which is developed to deal with data sparsity and limitation problems. The prediction from the decoder is based on the

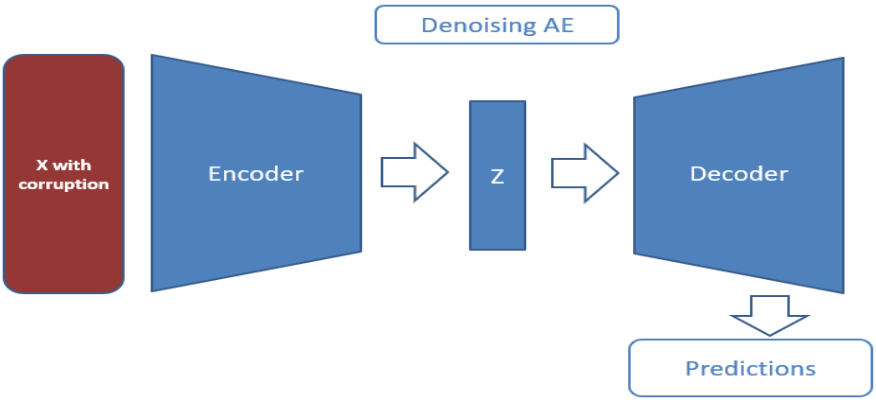

Figure 10: Model structure of denoising autoencoder.

latent variable z. In addition, to make the model more robust, it uses factitious noise on a raw input X, which is called corruption. The process of the decoder removing corruption from an input X makes the model more robust [36]. The learning objective of the model is to minimize reconstruction errors between raw data X and predicted values. Eventually, the autoencoder with convolutional layer presented better performance than the denoising autoencoder and the comparison between the two models will be provided in the experimental result.

3.1.3 Convolutinoal autoencoder

The overall structure of the proposed model is visualized in Figure 10. where the model consists of three main architectures : encoder, conditioing augmentation and decoder. The Encoder first encodes input data to a latent vector, Z. Conditioning augmentation (CA) extracts mean and variance of corresponding latent vectors. In addition, by using mean and variance, CA samples new latent vector Z'. Lastly, the decoder takes Z'

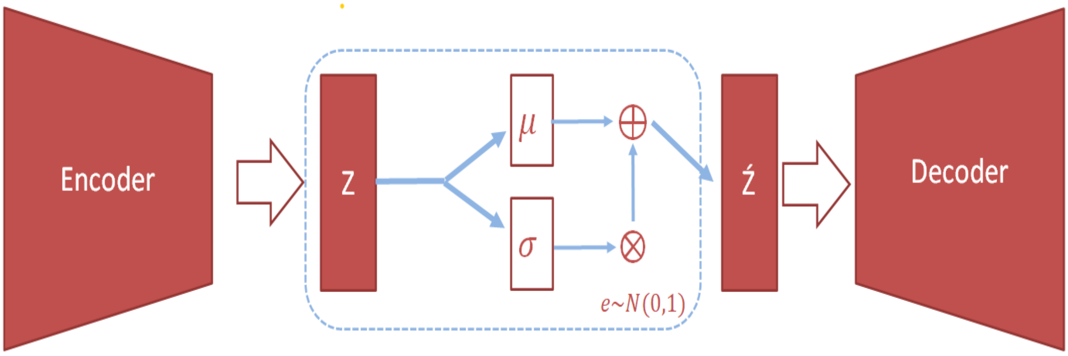

Figure 11: Model structure; proposed model includes three parts which are encoder, conditioning augmentation and decoder.

as input to output proper prediction.

3.1.4 Convolutional block

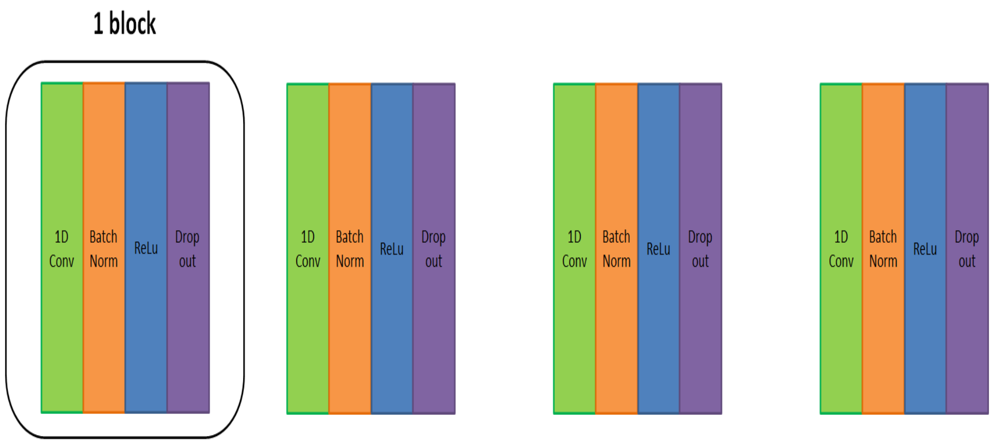

Figure 12: Convolutional blcok with four layers, convolutional layer, batchnormalization, activation and dropout

As mentioned above, the encoder and decoder are composed of convolutional blocks which contain convolutional layers, batch normalization, activation function which is a rectified linear unit, and drop out to deal with overfitting. The type of activation fucntion acts an important role in terms of the performance of the model [1], which is why it needs deliberation on choosing one. Most parameters, including the number of blocks, type of activation function, and proportion of dropout can be modified manually like hyper parameters.

3.1.5 Conditioning autmentation

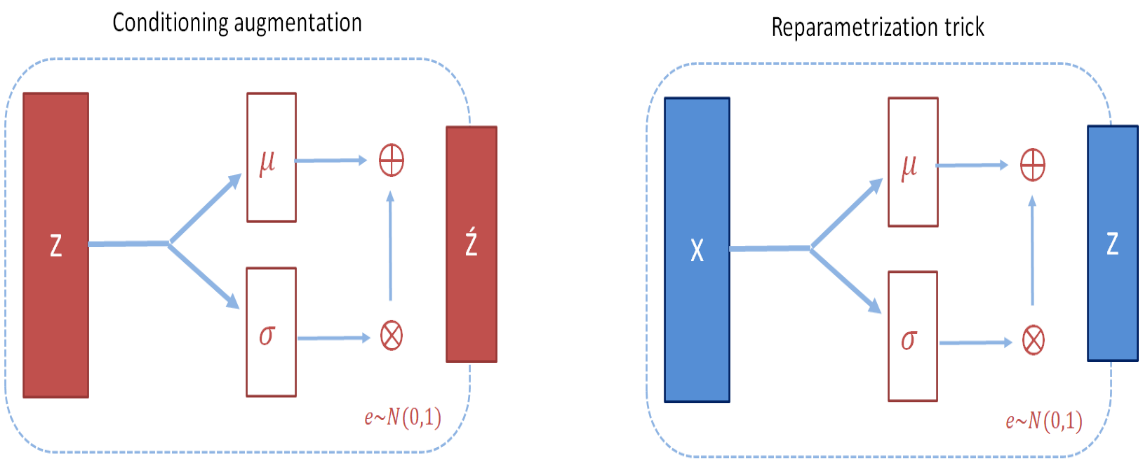

Figure 13: Comparison between conditioning augmentation and reparametrization trick

Conditioning augmentation (CA) which is the main concept to address the data limitation problem, was first introduced in the paper of stackGAN (2017). StackGAN is the model to generate image samples that correspond to text description. However, main challenge on this task is that it needs sufficient samples that are composed of image and text pair. Collecting or generating such datasets requires too much cost which is why StackGAN applied conditioning augmentation to generate additional text and image pair samples [37]. The process of conditioning augmentation resembles the process of reparametrization trick which is used in variational autoencoder to tackle backpropagation problems [37] []. However, these methods differ from two perspectives.

- Shape of input; As in Figure 12, conditioning augmentation takes input as latent vector Z, while the reparametrization trick takes input as X. In short, CA predicts mean and variance of latent vector Z and the reparametrization trick predicts mean and variance directly from raw input X.
- Purpose of method; From the perspective of purpose, CA was developed to deal with the lack of (text, image) pair samples which can be inferred from the name, while the reparametrization trick was designed to solve the problem of backpropagation that means directly Gaussian sampling from mean and variance makes the model impossible to implement backpropagation [38].

3.1.6 Comparison between denoising autoencoder and convolutional autoencoder with conditioning augmentation

As mentioned above, denoising autoencoder uses techniques of adding factitious noise on raw input X, while a conditioning augmentation resamples latent vector z using mean and variance predicted by fully connected layers [36]. It seemed conditioning augmentation presents better performance on dealing with data sparsity and limitation problem [37]. There can be various reasons for the low performance on denoising autoencoder and we assume the main reason for the result is that adding factitious mechanism on discrete variables such as ratings on user-item matrix. Denoisnig autoencoder usually show compliant performance when the input is composed of continuous variables.

3.1.7 Application on contents-based filtering recommendation

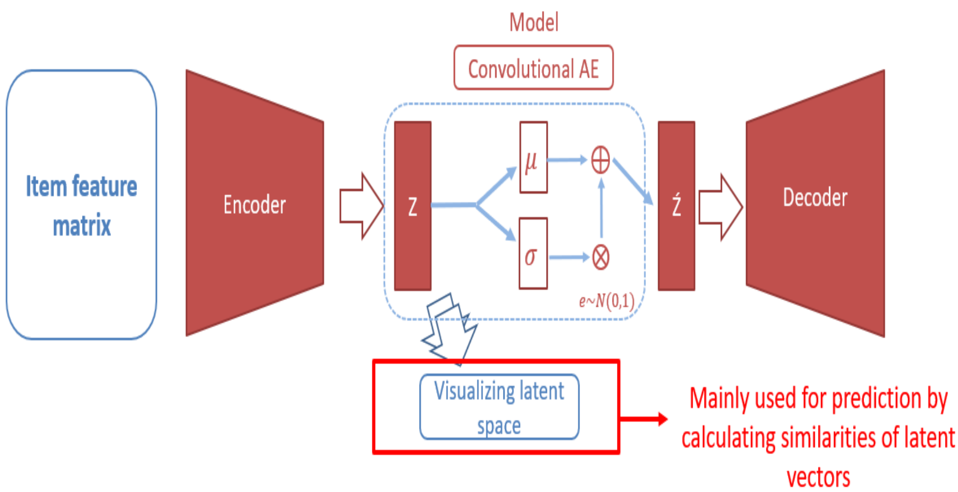

Figure 14: Prediction process of contents-based filtering of the proposed model

When the proposed model is applied to contents-based filtering recommendation. Prediction of the recommendation list is based on the latent space of embedded vectors z which is the output from the encoder. Each embedded vector indicates each item and the recommendation is implemented based on the similarities between embedded vectors. As shown on the Figure 15, a recommendation list is provided based on the similarities between a picked item and the others.

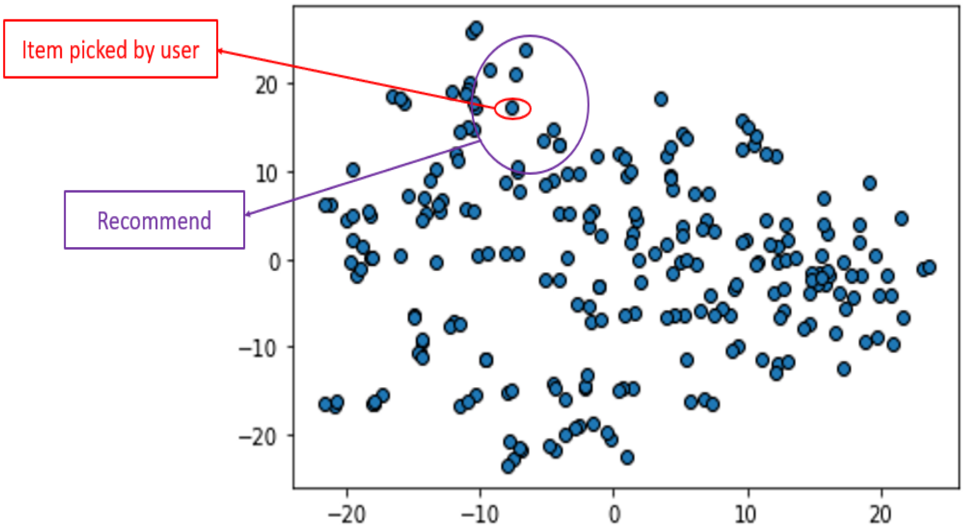

Figure 15: Visualization of a latent space of embedded vectors. Each embedded vector indicate each item

3.1.8 Application on collaborative filtering recommendation

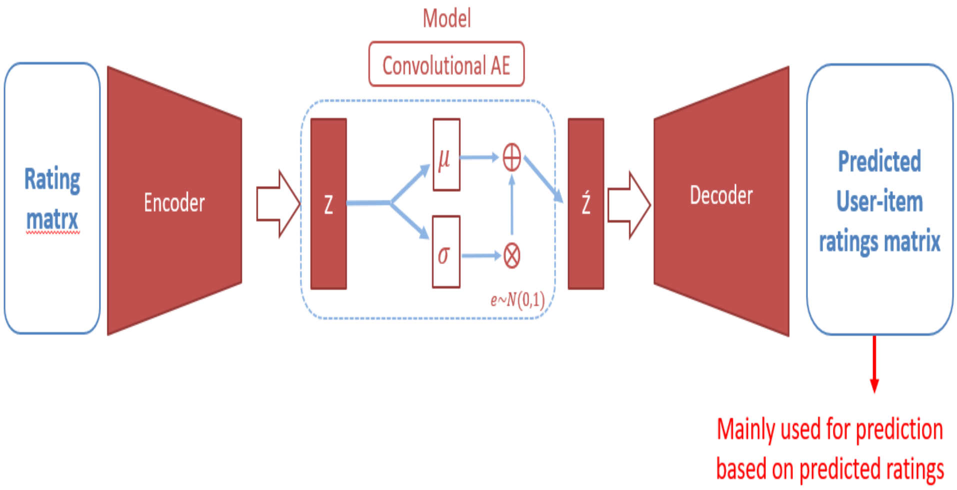

Figure 16: Prediction process of collaborative filtering of the proposed model

When the proposed model is applied to collaborative filtering recommendation. Ratings on user-item matrix is used as an input of the model. As the model learns representation from the matrix, decoder will be able to predict ratings of items of users which is mainly used for implementing recommendation. In short, main resource of contents-based filtering is an output from encoder, while main resource of collaborative filtering is an output from decoder. Evaluation of the proposed model is based on this approach, because this task provides an objective evaluation metric for comparison between the proposed model and other baselines.

3.2 Experimental result and analysis

This section presents the performance of the model to verify the proposed thesis compared to various base models that have shown high performance on corresponding benchmark dataset, Movielens, Netflix, and Flixster monti [39] [40] [41]. Base models are selected based on the performance of each dataset. Moreover, on the contetns-based filtering application, information of tourist place on Busan to represent item features of each place, collected from croud-sourcing platform was mainly used for visualization.

3.2.1 Datasets

Recommendation systems have to be robust to various types of dataset and also to the size of datasets. Movielens and Netflix open datasets are selected, because of the large size of the training set, which is the general case for major companies. In addition, the Flixster monti dataset is the case of a minor company that does not have a large pool of dataset. Moreover,

Movielens is a stable benchmark dataset containing about 20 million movie ratings by 162,000 users. Moreover, the dataset provides tagging applications for 4,000 movies, which is about a description of each movie written by users. Each rating is on a scale from 1 to 5

Netflix prize is the second most used open dataset in the recommender system field, containing 10 million movie ratings by users on 17,000 movies.

The dataset is also based on a 1 to 5 rating scale

Flixster monti is a classical recommendation bench mark dataset containing about 27,000 ratings for 3,000 different types of movies. Compared to the above two datasets, it has a smaller data pool with fewer customers. Also, the dataset includes 19 categories that indicate the genre for each movie.

In the case of Movielens and Netflix, experiments have been implemented on decreasing the size of each dataset, from 100 percent to 0.025 percent to verify the influence of conditioning augmentation on robustness of the proposed model compared to other base models.

3.2.2 Model parameters

Considering computing resources and the size of data to be handled, experimental settings for the model are the same which are determined by gridsearch that is frequently used for hyper parameter optimization in typical machine learning tasks. Most of the codes are implemented on the Pytorch framework and other detail settings will be provided in the appendix.

- Number of convolutional blocks: 4
- Activation function: Rectified linear unit
- Number of training epoch: 10
- Train/test/valid split: 6/2/2

- Batch size: 64
- Optimization: Adam optimizer

3.2.3 Evaluation metric

The proposed model is evaluated in the same way as other based models are evaluated on their bench mark datasets, which are rooted mean squared error (RMSE) for Flixster monti and Recall@K, normalized discounted cumulative gain (nDCG@K) for others. These are the most frequently used evaluation metric for recommender systems by using comparison between recommended lists and ideal recommendation lists [42]. Each measure can be defined as the following equations.

- Recall@K; Recall@K can be described as the number of relevant items among recommended items. Therefore, the term @K indicates the number of items on one list. The calculation of corresponding metric is the following.

$$Recall@k = \frac{\left|\hat{R}_k \cap R_k\right|}{R_k} \tag{2}$$

- nDCG@K; nDCG@K is more complicated metric than the others. It is the measure that compares the quality of recommended lists provided by the system and that of ideal recommendation lists. Moreover, this metric considers the order of the items on the recommended list which is why discounting factor for each item increases as the item is recommended on the behind order. Cumulative gain (CG) represents the sum of ratings of recommended list while, discounted cumulative gain (DCG) indicates the sum of discounted values of ratings. Lastly, ideal discounted cumulative gain (IDCG) is the DCG of an ideal recommendation list. Final measurement of nDCG@K can be calculated by dividing IDCG to DCG.

$$CG = \sum_{i}^{k} R_i, where\, R_i = Ratings\, of\, an\, items \tag{3}$$

$$DCG = \sum_{i}^{k} \frac{R_i}{log(1+i)} \tag{4}$$

$$IDCG = DCG(\grave{R}), where\, \grave{R} = Ratings\, of\, an\, ideal\, list \tag{5}$$

$$nDCG@N = \frac{DCG@k}{IDCG@k} \tag{6}$$

- RMSE; RMSE is one of the most frequently used metric for regression problem. On this task, RMSE is applied to the comparison between predicted ratings and true ratings of items.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{\sigma_i}\right)^2} \tag{7}$$

3.3 Experimental result and analysis

Base models for evaluation on two datasets (Movielens and Netflix) are high-ranked models including state of the art for each benchmark. H+ VAMP (2019) and VASP (2021) which are autoencoder based models, ranked high performance on Movielens, while H+ VAMP (2019) and Reccurrent variational autoencoder (RecVAE, 2019) ranked high performance on Netflix. Moreover, IGMC and sRGCNN which are based on graph neural network, showed high performance on Flixster monti. Movielens and Netflix evaluations are focused on the influence of the size of the datset, while Flixster monti focused on performance of the proposed model on the limited size of dataset situation. Recall@20, Recall@50, and nDCG@100 are chosen for evaluation metrics for Movielens and Netflix, while RMSE for Flixster monti.

3.3.1 Movielens and Netflix

As shown in Table 1 to 4, performance of the proposed model seems slightly lower than that of the state of the art with sufficient dataset size from 20,000,000 samples to 500,000. However, as the size of dataset decreases, the performance gap between the best model and the proposed model becomes smaller which means the proposed model with conditioning augmentation is more robust to the size of dataset. In addition, when the size of the dataset is less or equal to 100,000 samples, the proposed model showed higher performance than the state of the art. Specifically, about 3 to 4 percent of improvements have been made by the proposed model compared to the state of the art

Moreover, looking at Tables 1 and 2, the performance gap between designing with CA and without CA seemed significant suggesting CA helps to improve performance on learning representation from the inputs.

Table 1: Table 1: Movielens 20M, 100, 10, 5 percent of training set

Model	Recall@20	Recall@50	nDCG@100						
H+ VAMP	0.413	0.551	0.445	0.402	0.521	0.413	0.403	0.521	0.406
VASP	0.414	0.552	0.448	0.395	0.512	0.407	0.389	0.508	0.402
IGMC	0.398	0.503	0.425	0.382	0.498	0.411	0.38	0.491	0.405
sRGCNN	0.387	0.476	0.421	0.379	0.452	0.409	0.377	0.447	0.365
Proposed model w/o CA	0.378	0.475	0.415	0.377	0.456	0.408	0.369	0.445	0.374
Proposed model w/ CA	**0.402**	**0.513**	**0.421**	**0.4**	**0.511**	**0.416**	**0.399**	**0.487**	**0.409**
Dataset size	100% (20,000,000)			10% (2,000,000)			5% (1,000,000)		

Table 2: Table 2: Movielens 20M, 0.5, 0.375, 0.25 percent of training set

Model	Recall@20	Recall@50	nDCG@100						
H+ VAMP	0.391	0.425	0.401	0.391	0.421	0.4	0.389	0.414	0.399
VASP	0.383	0.421	0.398	0.381	0.418	0.395	0.381	0.415	0.387
IGMC	0.366	0.41	0.394	0.365	0.405	0.389	0.346	0.384	0.351
sRGCNN	0.367	0.437	0.357	0.35	0.43	0.351	0.339	0.4	0.346
Proposed model w/o CA	0.366	0.44	0.364	0.351	0.428	0.361	0.34	0.395	0.339
Proposed model w/ CA	**0.393**	**0.433**	**0.395**	**0.393**	**0.431**	**0.394**	**0.391**	**0.43**	**0.4**
Dataset size	0.5% (100,000)			0.375% (75,000)			0.25% (50,000)		

Table 3: Table 3: Netflix, 100, 10, 5 percent of training set

Model	Recall@20	Recall@50	nDCG@100						
H+ VAMP	0.376	0.462	0.409	0.35	0.441	0.385	0.348	0.442	0.349
RecVAE	0.361	0.452	0.394	0.348	0.437	0.36	0.345	0.431	0.359
IGMC	0.345	0.442	0.389	0.344	0.438	0.37	0.344	0.436	0.364
sRGCNN	0.336	0.439	0.387	0.334	0.428	0.375	0.331	0.419	0.34
Proposed model w/ CA	**0.359**	**0.448**	**0.393**	**0.349**	**0.435**	**0.388**	**0.348**	**0.433**	**0.386**
Dataset size	100% (10,000,000)			10% (1,000,000)			5% (500,000)		

Table 4: Table 4: Netflix, 0.1, 0.075, 0.05 percent of training set

Model	Recall@20	Recall@50	nDCG@100						
H+ VAMP	0.341	0.437	0.373	0.337	0.423	0.368	0.338	0.415	0.366
RecVAE	0.339	0.426	0.355	0.33	0.421	0.349	0.321	0.422	0.341
IGMC	0.327	0.42	0.347	0.325	0.41	0.342	0.315	0.405	0.335
sRGCNN	0.321	0.4	0.334	0.318	0.398	0.329	0.314	0.387	0.315
Proposed model w/ CA	**0.342**	**0.433**	**0.375**	**0.339**	**0.431**	**0.374**	**0.343**	**0.429**	**0.369**
Dataset size	0.1% (100,000)			0.075% (75,000)			0.05% (50,000)		

3.3.2 Flixster monti

On Flixster monti, the proposed model ranked second performance, meaning it shows compliant performance on various types of dataset even with limited size of dataset.

Model	RMSE
GRALS	1.245
sRGCNN	0.926
VASP	0.889
H+VAMP	0.885
IGMC	0.872
Proposed model w/ CA	**0.879**

Table 5: Flixster monti, 26,173 samples

3.3.3 Overall comparison

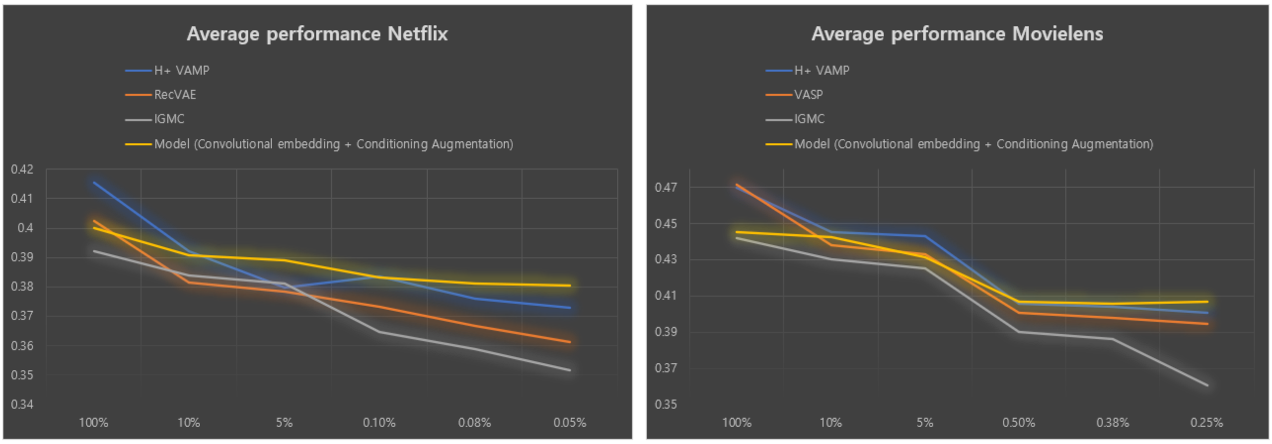

Figure 17: Comparison of average performance of Recall@20, Recall@50 and nDCG@100

The proposed model with conditioning augmentation has proved to be competitive with diverse sizes of the dataset, while most of the base models presented high performance with highly conditioned circumstances such as sufficient size of samples or specific types of data. On average performance of three main metrics, Recall@20, Recall@50, and nDCG@100, Figure 13 presents the robustness of the proposed model to the size of datset.

4 Conclusion

We have presented an autoencoder-based recommender system with a convolutional layer that can be applied to both collaborative filtering and content-based recommending tasks. Moreover, by combining conditioning augmentation technique with convolutional autoencoder, we tried to design a model that is robust to the size of the dataset and the experimental results have shown that applying this technique leads to better performance in specific circumstances.

However, one of the limitations is that the proposed model does not show great performance when sufficient training samples are prepared. Reasons for this limitation can be diverse, such as model structure, parameter optimization, or preprocessing and improving this limitation may stay as a future work.

Lastly, another important finding is that the shape and type of input source of recommender systems are not limited to rating. Recently, many users are uploading their lifestyles or preferences on their social media platforms or crowdsourcing platforms by taking pictures, writing reviews, and even creating content like taking videos. Thereafter, learning representation from unstructured data such as images and texts became more important than ever [43], which is why we started to have an interest in text-to-image or image-to-text transforming tasks for future work, example provided in Figure 14. Learning features from image-to-text or vice versa may provide plenty of resources for recommender systems, especially in the Korean language, while research on Korean language on corresponding tasks has not been studied deeply yet.

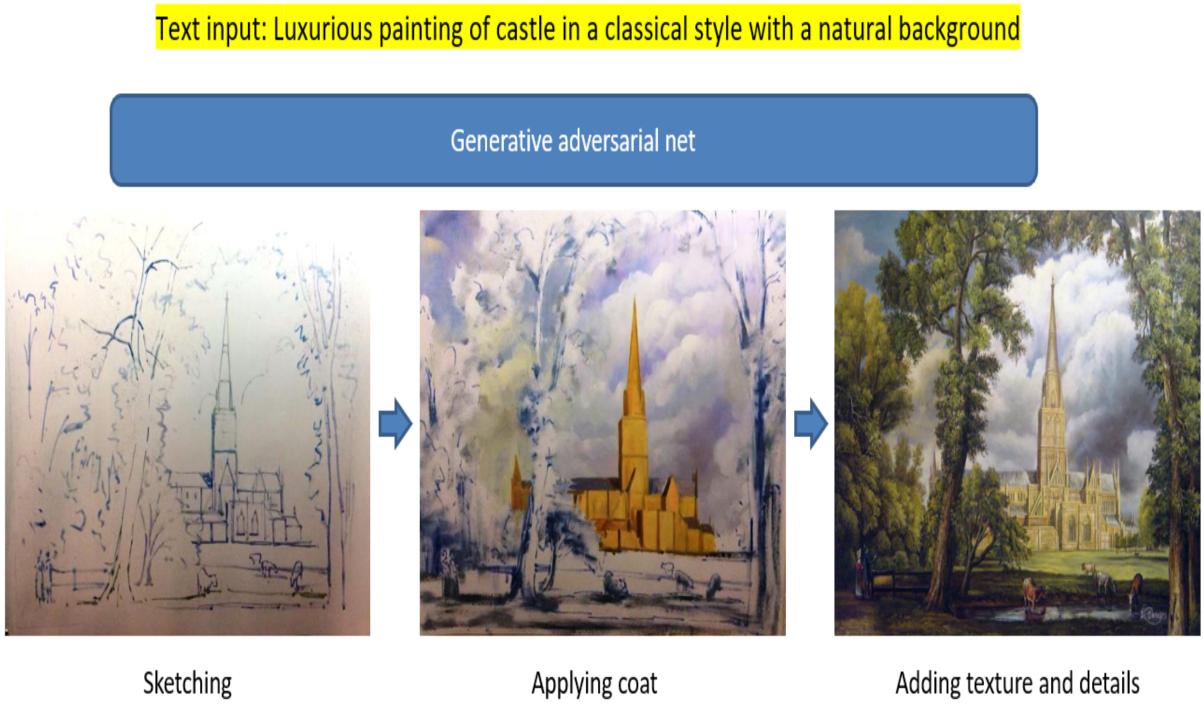

Figure 18: Text to image generating example by analogy of human painting

References

[1] B. Bai, Y. Fan, W. Tan, and J. Zhang, "Dltsr: a deep learning framework for recommendations of long-tail web services," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 73–85, 2017.

[2] S. Cao, N. Yang, and Z. Liu, "Online news recommender based on stacked auto-encoder," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. IEEE, 2017, pp. 721–726.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[4] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[5] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian informatics journal*, vol. 16, no. 3, pp. 261–273, 2015.

[6] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-based systems*, vol. 46, pp. 109–132, 2013.

[7] B. T. Betru, C. A. Onana, and B. Batchakui, "Deep learning methods on recommender system: A survey of state-of-the-art," *International Journal of Computer Applications*, vol. 162, no. 10, pp. 17–22, 2017.

[8] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and trends® in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[9] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[10] A. Singhal, P. Sinha, and R. Pant, "Use of deep learning in modern recommendation system: A summary of recent works," *arXiv preprint arXiv:1712.07525*, 2017.

[11] F. S. de Aguiar Neto, A. F. da Costa, M. G. Manzato, and R. J. Campello, "Preprocessing approaches for collaborative filtering based on hierarchical clustering," *Information Sciences*, vol. 534, pp. 172–191, 2020.

[12] W. Cetera, W. Gogołek, A. Żołnierski, and D. Jaruga, "Potential for the use of large unstructured data resources by public innovation support institutions," *Journal of Big Data*, vol. 9, no. 1, pp. 1–21, 2022.

[13] L. Zhang, Z. Qi, and F. Meng, "A review on the construction of business intelligence system based on unstructured image data," *Procedia Computer Science*, vol. 199, pp. 392–398, 2022.

[14] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[15] S. Kanwal, S. Nawaz, M. K. Malik, and Z. Nawaz, "A review of text-based recommendation systems," *IEEE Access*, vol. 9, pp. 31 638–31 661, 2021.

[16] D. Jannach, A. Manzoor, W. Cai, and L. Chen, "A survey on conversational recommender systems," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021.

[17] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[18] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston *et al.*, "The youtube video recommendation system," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 293–296.

[19] K. K. Pal and K. Sudeep, "Preprocessing for image classification by convolutional neural networks," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, 2016, pp. 1778–1781.

[20] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Analytics*, vol. 1, no. 1, pp. 1–22, 2016.

[21] M. D. D. Mortha, S. Maddala, and V. Raju, "Data preprocessing for learning, analyzing and detecting scene text video based on rotational gradient," in *International Conference on Data Science, E-learning and Information Systems 2021*, 2021, pp. 1–8.

[22] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional

networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2272–2281.

[23] M. Elahi, F. Ricci, and N. Rubens, "A survey of active learning in collaborative filtering recommender systems," *Computer Science Review*, vol. 20, pp. 29–50, 2016.

[24] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, 2009.

[25] J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks," *Expert Systems with Applications*, vol. 89, pp. 404–412, 2017.

[26] M. Vanetti, E. Binaghi, B. Carminati, M. Carullo, and E. Ferrari, "Content-based filtering in on-line social networks," in *International Workshop on Privacy and Security Issues in Data Mining and Machine Learning.* Springer, 2010, pp. 127–140.

[27] P. B. Thorat, R. M. Goudar, and S. Barve, "Survey on collaborative filtering, content-based filtering and hybrid recommendation system," *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31–36, 2015.

[28] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.

[29] M. Karimi, D. Jannach, and M. Jugovac, "News recommender systems–survey and roads ahead," *Information Processing & Management*, vol. 54, no. 6, pp. 1203–1227, 2018.

[30] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2015.

[31] P. Lops, M. d. Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," *Recommender systems handbook*, pp. 73–105, 2011.

[32] S. U. Tareq, M. Noor, C. Bepery *et al.*, "Framework of dynamic recommendation system for e-shopping," *International Journal of Information Technology*, vol. 12, no. 1, pp. 135–140, 2020.

[33] X. Cai, J. Han, and L. Yang, "Generative adversarial network based heterogeneous bibliographic network representation for personalized citation recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[34] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.

[35] O. Kuchaiev and B. Ginsburg, "Training deep autoencoders for collaborative filtering," *arXiv preprint arXiv:1708.01715*, 2017.

[36] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[37] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915.

[38] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[39] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[40] J. Bennett, S. Lanning *et al.*, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007. New York, 2007, p. 35.

[41] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[42] F. H. Del Olmo and E. Gaudioso, "Evaluation of recommender systems: A new approach," *Expert Systems with Applications*, vol. 35, no. 3, pp. 790–804, 2008.

[43] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International conference on machine learning*. PMLR, 2016, pp. 1060–1069.

[44] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 565–573.

[45] M. Unger, "Latent context-aware recommender systems," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 383–386.

[46] F. Strub, R. Gaudel, and J. Mary, "Hybrid recommender system based on autoencoders," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 11–16.

[47] M. Naumov, D. Mudigere, H.-J. M. Shi, J. Huang, N. Sundaraman, J. Park, X. Wang, U. Gupta, C.-J. Wu, A. G. Azzolini *et al.*, "Deep learning recommendation model for personalization and recommendation systems," *arXiv preprint arXiv:1906.00091*, 2019.

[48] T. Ebesu and Y. Fang, "Neural citation network for context-aware citation recommendation," in *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, 2017, pp. 1093–1096.

[49] T. Alashkar, S. Jiang, S. Wang, and Y. Fu, "Examples-rules guided deep neural network for makeup recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[50] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 355–364.

[51] Y. Zhao, K. Wang, G. Guo, and X. Wang, "Learning compact yet accurate generative adversarial networks for recommender systems," *Knowledge-Based Systems*, vol. 257, p. 109900, 2022.

[52] H. Bharadhwaj, H. Park, and B. Y. Lim, "Recgan: recurrent generative adversarial networks for recommendation systems," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 372–376.

[53] M. Gao, J. Zhang, J. Yu, J. Li, J. Wen, and Q. Xiong, "Recommender systems based on generative adversarial networks: A problem-driven perspective," *Information Sciences*, vol. 546, pp. 1166–1185, 2021.

[54] A. Prosvetov, "Gan for recommendation system," in *Journal of Physics: Conference Series*, vol. 1405, no. 1. IOP Publishing, 2019, p. 012005.

[55] D. Yang, J. Zhang, S. Wang, and X. Zhang, "A time-aware cnn-based personalized recommender system," *Complexity*, vol. 2019, 2019.

[56] H.-w. An and N. Moon, "Design of recommendation system for tourist spot using sentiment analysis based on cnn-lstm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 3, pp. 1653–1663, 2022.

[57] D. Kim, C. Park, J. Oh, and H. Yu, "Deep hybrid recommender systems via exploiting document context and statistics of items," *Information Sciences*, vol. 417, pp. 72–87, 2017.

[58] R. He and J. McAuley, "Vbpr: visual bayesian personalized ranking from implicit feedback," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[59] L. Huang, M. Fu, F. Li, H. Qu, Y. Liu, and W. Chen, "A deep reinforcement learning based long-term recommender system," *Knowledge-Based Systems*, vol. 213, p. 106706, 2021.

[60] I. Munemasa, Y. Tomomatsu, K. Hayashi, and T. Takagi, "Deep reinforcement learning for recommender systems," in *2018 international conference on information and communications technology (icoiact)*. IEEE, 2018, pp. 226–233.

[61] L. Zou, L. Xia, Z. Ding, J. Song, W. Liu, and D. Yin, "Reinforcement learning to optimize long-term user engagement in recommender systems," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2810–2818.

[62] C. Lei, D. Liu, W. Li, Z.-J. Zha, and H. Li, "Comparative deep learning of hybrid representations for image recommendations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2545–2553.

5 Appendix

5.1 Appendix A: Preprocessing of text data for visualization of a latent space

The preprocessing of textual data crawled from the croud-sourcing platform was based on a basic natural language process technique called term frequency matrix. The term frequency matrix technique is similar to the bag of word method that considers the importance of each word based on the frequency of occurrence of each word. Moreover, to utilize the attributes of the Korean language, the words on the vector are arranged in the order of Korean sentence structure and the number of each part of speech is set to 40 60 like in Figure 19.

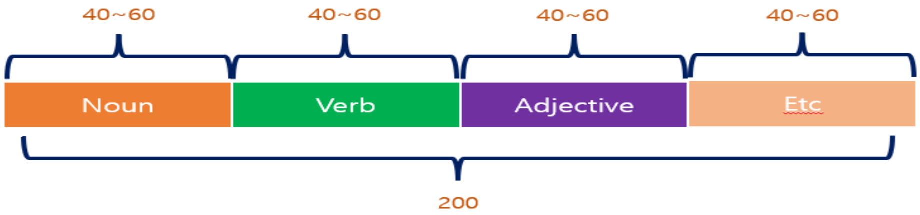

Figure 19: Structure of processed vectors of Korean sentences

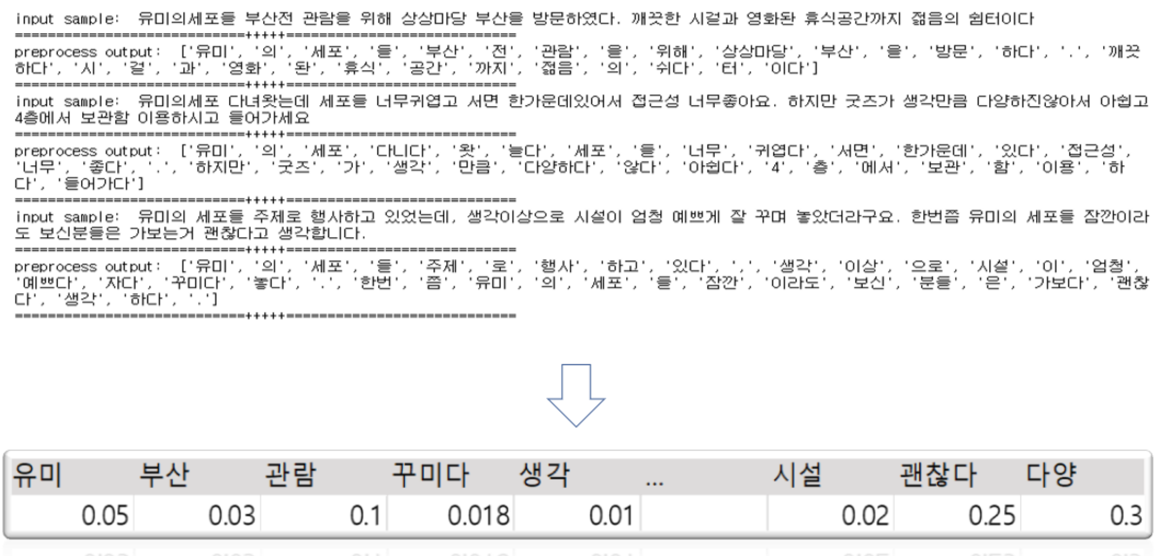

Figure 20: Transformation from text to vectors

5.2 Appendix B: Table of paper; deep learning for recommender system

Table of publications related to deep learning technique applied to recommender systems. One publication can belong to several model categories if it uses combined type of model.

Table 6: Table of publications related to deep learning based recommender system

Model	Publications
Autoencoder	[11], [23], [44], [45], [39], [46], [1], [7], [34]
Multi layer perceptron	[47], [18], [40], [48], [49], [50]
Generative adversarial network	[51], [52], [53], [54], [33]
Convolutional neural network	[12], [7], [30], [55], [56], [57], [58]
Reinforcement learning	[59], [60], [61]
Hybrid	[44], [46], [62], [9]