An Investigation into Negative Sampling of Two-Tower Neural Networks for Large Corpus Item Recommendations

Sivananda Ganesamoorthy

Master of Science in Computer Science
The University of Bath
August 2024

An Investigation into Negative Sampling of Two-Tower Neural Networks for Large Corpus Item Recommendations

Submitted by: Sivananda Ganesamoorthy

Copyright

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

Recommendation systems, employed in many industry verticals such as e-commerce, media content and advertisement, are a much required commercial tool in today's Online world. With the acceleration of the internet over the last 2 decades, today's recommendation systems need to be able to score, rank and retrieve items from a very large corpus. The Two Tower architecture, a very popular deep neural network (DNN) that powers some of the world's busiest recommendation engines typically encodes the user-related information in one tower and item information in the other tower before performing a dot product between the two towers; the goal being to recommend items based on a user's past item interactions. One of the biggest challenges that remains in the Two Tower model is understanding items that a user did not interact with and ensuring there is no selection bias, particularly from implicity user feedback, from all of the positive labels. Negative sampling helps combat this challenge as negative interactions (user-non-item pairs) are randomly sampled from a subset of all user-item pairs. This dissertation investigates existing Negative Sampling techniques, in particular, in-batch, mixed-batch and cross-batch sampling and the challenges that they overcome from existing literature. Experiments are conducted for each technique with a variety of loss criteria such as decoupled-softmax and Top-K and metrics such as Precision@k, Recall@k and nDSG@k are recorded. The final results are evaluated and the paper concludes with providing a direction of negative sampling strategies for future research.

Acknowledgements

I am deeply grateful to Dr. Rohit Babbar from the Department of Computer Science for his invaluable guidance and support throughout this dissertation. His insights and recommendations have significantly influenced the direction of my research and motivated me to achieve a high standard of work, especially especially given my initial limited knowledge on the subject.

I would also like to express my deepest gratitude to my parents. Their instilled educational values have been instrumental in my decision to revisit my academic journey 18 years into my career.

Finally, I must extend my heartfelt thanks to my loving wife, Sylvia, and my wonderful children, Shreya and Ethan, for their unwavering support and patience throughout my Master's programme. The completion of this dissertation would not have been possible without their encouragement and understanding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last decade, recommendation systems have played a crucial commercial role in increasing user engagement across internet-based businesses. Product catalogs such as Amazon, Video catalogs such as Youtube and Netflix, Search Engines and Games are just some categories where recommendation systems exist to ensure that users remain engaged. Often, a recommendation starts with the notion of "user A engaged with this item and so user A may also engage with these similar items" (item to item similarities) or "user A and user B engaged with similar products; user A engaged with item A so user B may also engage with item A".

Today, recommendation systems need to be able to scale to millions and billions of labels in very low latency conditions.

1.1 Example Section

Like all chapters, it will have a number of sections ...

1.1.1 Example Subsection

... and subsections ...

Example Sub-subsection

... and sub-subsections.

1.2 Another Section With a Long Title and Whose Title Is Abbreviated in the Table of Contents

Table 1: An example table

Items	Values
Item 1	Value 1
Item 2	Value 2

Another section, just for good measure. You can reference a table, figure or equation using `\ref`, just like this reference to Table 1.

1.3 Example Lists

1.3.1 Enumerated

1. Example enumerated list:

    - a nested enumerated list item;
    - and another one.
2. Second item in the list.

1.3.2 Itemised

- Example itemised list.
    - A nested itemised list item.
- Second item in the list.

1.3.3 Description

Item 1 First item in the list.

Item 2 Second item in the list.

Chapter 2

Literature Review

Recommendation systems are integral to modern digital ecosystems, enabling users to navigate the overwhelming volume of available content and items. They are widely used across various domains, including e-commerce, streaming services, social media, and the Internet of Things. The primary goal of these systems is to enhance the user experience by providing personalised suggestions based on a variety of data sources.

Benefits of Recommendation Systems

Recommendation systems offer significant benefits to users. They manage information overload by filtering and prioritising options, helping users navigate the vast array of choices in today's digital landscape. This makes it easier for users to find what they are looking for. Additionally, these systems personalise suggestions to individual users, making for a more engaging and relevant experience, which can increase user satisfaction and loyalty. For instance, in the context of the Internet of Things, Felfernig et al. (2019) highlights that recommendation technologies support the efficient identification of relevant artifacts, making them key technologies of future solutions.

Recommendation systems also leverage user data, such as historical interactions and preferences, to iteratively improve recommendations. Interestingly, Melchiorre et al. (2021) found that the Bayesian Personalised Ranking, an algorithm that optimises recommendations based on user-item interaction data, returned the most set of diverse recommendations in their experiments despite having weaker accuracy metrics.

Effective recommendation systems can also drive sales by suggesting complementary products or services, thereby increasing the average order value. They encourage users to spend more time on platforms by presenting them with content that aligns with their interests. Furthermore, by providing relevant suggestions, recommendation systems improve the overall user experience, leading to higher retention rates and positive feedback.

Types of Recommendation Approaches

Recommendation approaches may be categorised into several types. Collaborative filtering recommends items based on the preferences of similar users. It assumes that users who have agreed in the past will agree in the future (Aggarwal, 2016; Gupta and Dave, 2020; Panda and Ray, 2022). This approach can be further divided into user-based collaborative filtering, which suggests items based on the preferences of similar users, item-based collaborative filtering, which recommends items similar to those the user has liked in the past and latent-factor based matrix factorization, which recommends items based on latent features of both items and users (Aggarwal, 2016; Panda and Ray, 2022).

Content-based filtering, on the other hand, recommends items by comparing the content of items that a user has liked in the past with new items. For example, if a user enjoys a particular genre of music, the system will suggest similar genres (Aggarwal, 2016; Felfernig et al., 2019; Panda and Ray, 2022).

Hybrid approaches combine collaborative filtering and content-based filtering to leverage the strengths of both methods while mitigating their weaknesses. For instance, a hybrid group recommendation model can consider both individual user preferences and group cohesion to enhance the recommendation quality (Aggarwal, 2016; Jeong and Kim, 2019).

Context-aware recommendations consider additional contextual information such as location, time, and user activity to tailor suggestions. This approach, particularly relevant in the Internet of Things, enhances the relevance of recommendations by considering the user's current situation (Felfernig et al., 2019). Aggarwal (2016) further observes that such contextual information greatly improves the effectiveness of the recommendation process.

Two-Tower Neural Networks

In today's digital landscape, recommendation systems need to be able to recommend items from large corpora which has led to the increasing importance of deep learning neural networks. Deep learning neural networks are able to model more complex patterns in data, achieve much higher accuracy and offer flexibility in design (Schifferer, 2021) then traditional approaches such as matrix factorization or collaborative filtering. One such deep learning neural network is the two-tower neural network, often referred to as a dual-encoder architecture.

Two-tower neural networks are a popular framework in recommendation systems designed to learn separate representations for users and items. This architecture consists of two towers: one tower encodes user features and the other tower encodes item features. The outputs from these networks are then combined using a simple operation, such as dot product or cosine similarity, to generate a score that indicates the relevance of an item to a user, thus allowing for efficient retrieval of items from a large pool.

Negative Sampling

Negative sampling is a technique often employed in training two tower networks, especially when dealing with implicit feedback data, by selecting negative examples (items a user has not interacted with) alongside positive examples (items a user has interacted with). This approach allows the model to learn to distinguish between liked and disliked items effectively, thereby improving the predictive accuracy of the model. Negative sampling is particularly important to large corpora since it is efficient in training (Chen et al., 2020; Fan et al., 2023; Yang et al., 2024), handles imbalanced data (Chen et al., 2020, 2023; Yang et al., 2024), improves model performance (Ding et al., 2019; Yang et al., 2024) and mitigates selection bias in the model (Yang Google et al., 2020)

Having introduced recommendation systems, two-tower neural networks and negative sampling, this Literature Review aims to:

1. Describe and discuss the Two-Tower Architecture in the context of recommendation systems.
2. Review and discuss various negative sampling strategies
3. Discuss the current state-of-the-art negative sampling techniques within the context of the two-tower architecture.
4. Review metrics that are commonly used to evaluate the performance of two-tower

models.

5. Discuss some real world case studies where the two-tower model has been deployed with negative sampling for large corpus recommendations.

2.1 Two-Tower Neural Networks

2.1.1 Overview

The two-tower neural network, otherwise known as the dual-encoder architecture (Gupta et al., 2023), is a popular and widely used neural network in recommendation systems as it is designed to efficiently match users with items. As illustrated in Figure 1, the architecture consists of two distinct towers; a user tower for encoding user representations such as demographic data and previous interactions based on an input query and an item tower for encoding item representations such as titles and other item related metadata.

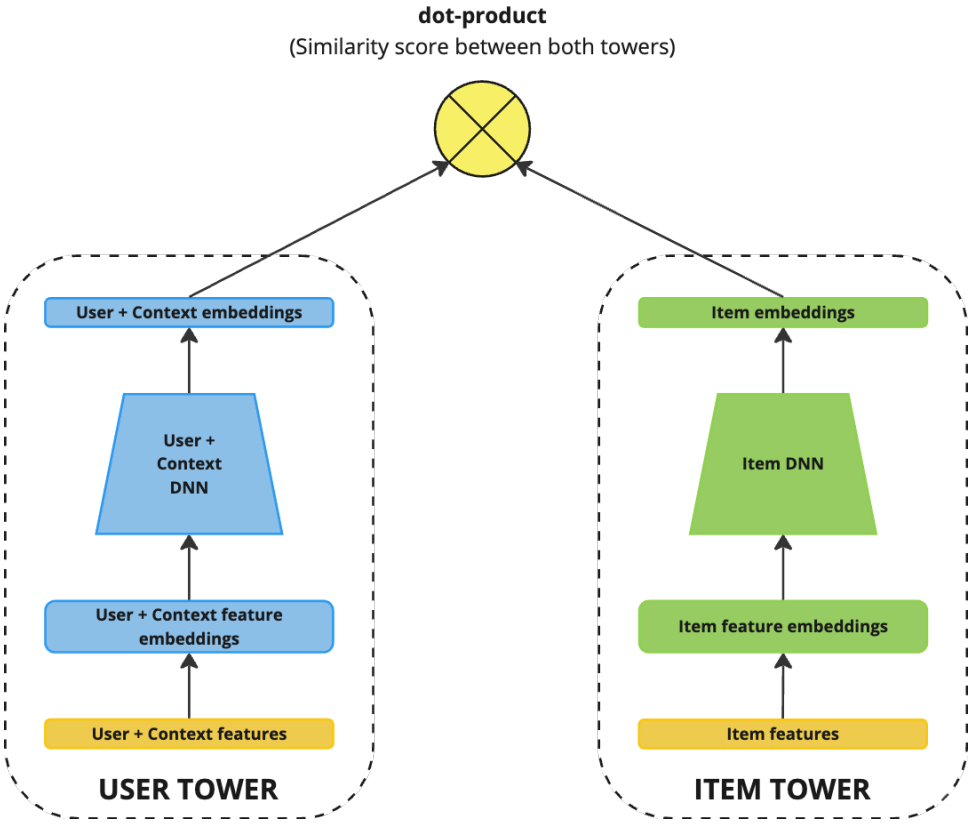

Figure 1: A dual-encoded two-tower architecture, adapted from (Li et al., 2022)

The dataset that is used in a two-tower architecture may be represented as a query set $\{u_i\}_{i=1}^{N}$ and an item set $\{v_j\}_{j=1}^{M}$ where N is the number of users and M is the number of items (Yu et al., 2021).

The inputs to the user and item towers, u_i and v_j respectively, are fed through embedding layers to obtain the user and item representations as vectors. Da, Kou and Peng (2022) used Long-Short-Term-Memory (LSTM) networks to encode the inputs in their proposal of a dual encoder retrieval model for citation recommendations. Citations tend to be keyword heavy

and in this instance LSTM, capable of handling sequential data, helped with the vanishing gradient problem.

However, for most other recommendation systems, one-hot encoding of the inputs is sufficient as this provides a generic vector representation of the latent features (He et al., 2017). For each tower, this results in feature embeddings $e = [e_1, e_2, ..., e_m]$ where $e_i \in \mathbb{R}_d$ is the embedding of the i-th feature and d is the embedding dimension (Li et al., 2022).

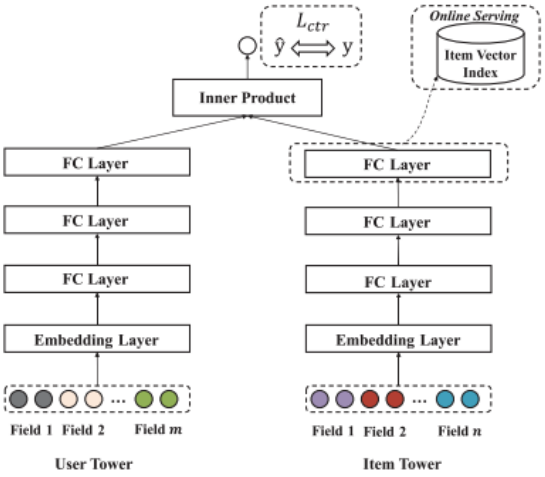

Figure 2: Overview of neural network based two-tower model (Li et al., 2022)

The Deep Neural Network (DNN) for both towers is a Multi Layer Perceptron (MLP) that includes some hidden layers as well as fully-connected layers as illustrated in Figure 2. To get the representations of the user and item towers, the fully-connected layers go through an L2 normalization layer resulting in $p_u = L2Norm(h^L)$ for the user representation and $p_v = L2Norm(h^L)$ for the item representation. L denotes the depth of the fully-connected layers h. The final output of the model is the dot product of the query and item embedding resulting in a matrix score, $s(u, v)$ (Yu et al., 2021; Li et al., 2022):

$$s(u, v) = \langle p_u, p_v \rangle \tag{1}$$

2.1.2 Benefits of the Two-tower Model

The two-tower model may be considered a hybrid model and offers several advantages over other architectures such has collaborative filtering and content-based filtering. The simplicity of the model's architecture, namely the two towers, allows for it to be easily understood and implemented for different recommendation system applications, compared to more complex architectures.

Another advantage is scalability. Given that the two towers are processed in a parallel manner, the architecture is able to scale to handle millions of items and queries, thus making it suitable for industrial applications. Additionally, since the architecture relies on pre-computing and caching the item embedding (Su et al., 2023), this significantly speeds up the retrieval process making it quite an efficient model.

Each tower is also able to leverage different latent user and item features making it a flexible architecture that is able to capture latent representations of both users and items. Importantly, this allows the model to better address the cold-start issue, where new users or new items lack sufficient historical interaction (Lee and Cho, 2023).

2.1.3 Challenges

Despite its advantages, the two-tower model comes with challenges that must be addressed. Firstly, the lack of interaction between the user and item towers during the training phase prevents the model from leveraging potential cross-information. This separation can hinder the model's ability to learn shared representations, potentially degrading its performance (Li et al., 2022; Shan et al., 2023; Su et al., 2023).

Secondly, the architecture suffers from data imbalance issues, particularly due to the sparsity of implicit feedback. Implicit feedback often contains disproportionately more positive interactions leading to a selection bias (Ding et al., 2019; Chen et al., 2023; Yang et al., 2024). This bias can cause the recommendation system to recommend items that are popular within the corpus but not relevant to the user's preferences or historical engagement patterns.

Finally, although the two-tower architecture mitigates the cold-start issue, it has been shown to under-perform in certain scenarios. As highlighted by Lee and Cho (2023) the two-tower model under performs on understanding user features when only the item-representations are shared inside the user-encoder.

To address these challenges, more complex solutions exist (Yu et al., 2021; Li et al., 2022). However, negative sampling stands out as a straightforward and computationally efficient method to enhance the model's performance.

2.2 Negative Sampling in Two-Tower Networks

The goal of negative sampling is to improve the efficiency and effectiveness of a model's training process. The model does this by taking a small subset of negative samples from the entire pool of possible negative samples (Yang et al., 2024). More formally, Yang et al. (2024) presents the mathematical definition of negative sampling as $L = l(x, x^+, x^-), x^- \sim p_n(x^-)$.

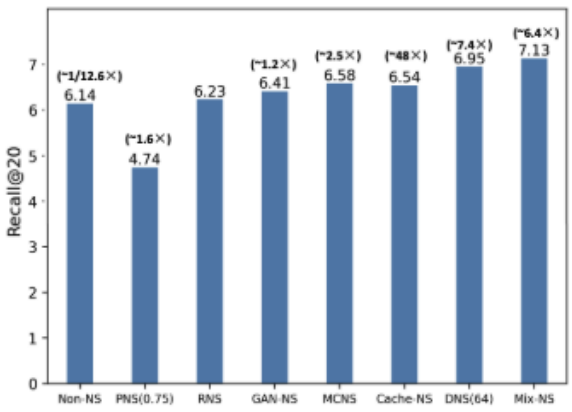

Figure 3: Performance and convergence speed comparison with various negative sampling methods (Yang et al., 2024)

From the sampling distribution $p_n(x^-)$, negative sampling would select a sample x^- from the pool of all sample candidates for a specific positive sample x^+ or anchor sample x, where $l(.)$ is a specific loss function such as Bayesian Personalized Ranking loss or InfoNCE loss.

Several negative sampling strategies exist in the context of two-tower networks to help solve the challenges of data imbalances, limited interaction between the towers and the cold-start issue. The earliest implementation of negative sampling is random negative sampling where a subset of negative samples from the corpus is randomly selected (Yang et al., 2024). Infact, Rendle et al. (2012) utilised random negative sampling in their proposed optimisation criterion of Bayesian Personalised Ranking to solve the lack of negatives in implicit feedback. However, while random negative sampling may be computationally

efficient, since only a subset of negative samples are selected, the model performance degrades on larger datasets as the relevancy of the selected negatives reduces during training (Yang et al., 2024).

The limitation of random negative sampling led to popularity-based negative sampling where more popular negative items in the corpus are selected on the assumption that they are likely to be true negatives if they frequently occur (Yang et al., 2024). Yang Google et al. (2020) used popularity-based negative sampling within each training batch in their proposal of mixed negative sampling to solve the selection bias issue.

Hard negative sampling strategies, such as dynamic negative sampling and mixture-based dynamic negative sampling later came about, focusing on the selection of challenging negative samples. These strategies involve identifying and utilising samples that the model struggles to classify, as opposed to random negative sampling or popularity-based negative sampling where easy negatives don't contribute towards the model's learning process. Dynamic negative sampling dynamically adjusts the selection of negative samples during training, targeting samples that are particularly difficult for the model to learn. This adaptability allows the model to continue to learn and improve its performance (Yang et al., 2024).

On the other hand, mixture-based dynamic negative sampling synthesizes negatives from a selected list of negative samples within the embedding space. This synthetic creation of negative samples allows the model to learn on really hard negatives. Figure 3 confirms both dynamic negative sampling and mixture-based dynamic negative sampling to have faster convergence and better performance over static counterparts, random negative sampling and popularity-based negative sampling. While dynamic negative sampling shows a better convergence when compared with mixture-based dynamic negative sampling, it suffers from a slightly worse performance.

2.2.1 State-of-the-art in Negative Sampling

Reinforced Negative Sampler

Inspired by Reinforced Learning, Ding et al. (2019) proposed a novel Reinforced Negative Sampler which integrates latent user information to generate high quality negatives. The objective of Reinforced Negative Sampler was to address the selection bias from implicit feedback. Primarily, the design of the Reinforced Negative Sampler involves an adversarial sampler, to generate hard negative instances, and an exposure-matching sampler that generates real negative instances based on the exposure data from users' interactions and non-interactions.

Theoretically, by being able to combine hard and real negative instances, Ding et al. (2019) will have reduced the amount of false negatives but this is not explicitly mentioned in their analysis. Scepticism should also be drawn towards the expected additional computational overhead of Reinforced Negative Sampler when used on an extremely large dataset with multiple latent features for both users and items; it is noted that their analysis was performed on a subset of data from potentially large dataset and click data, which is naturally limited in latent features.

Despite these contradictions, in their comparisons against Bayesian Personal Ranking, an earlier sampling approach that addressed Ranking issues on implicit feedback (Rendle et al., 2012), Reinforced Negative Sampling performed better than Bayesian Personalised Ranking methods on Area-Under-Curve (AUC) and Normalised Discounted Cumulative Gain (nDCG).

Mixed Negative Sampling

Yang Google et al. (2020) proposed a novel Mixed Negative Sampling approach combing both batch negatives and additional uniformly sampled negatives from the entire corpus to address the selection bias issue within two-tower networks. An advantage that Mixed Negative Sampling offers is the flexibility to control the sampling distribution by adjusting the batch from the additional sampled negatives via hyper-parameter tuning.

Mixed Negative Sampling was deployed to Google Play's app recommendation system confirming the scalability of this negative sampling technique. When experimented against batch negative sampling on the Google Play dataset, mixed negative sampling performed significantly better. Furthermore on live experiments, it was found that mixed negative sampling offered a 1.54% statistical significance increase in high-quality app installs, thus validating the efficiency and accuracy of this approach.

Cross-Batch Negative Sampling

Cross-Batch Negative Sampling differs from Mixed Negative Sampling in that instead of training on in-batch negative sampling, Cross-Batch Negative Sampling uses an ephemeral first-in-first-out (FIFO) memory bank to store previous item embeddings and use those across mini-batches (Wang, Zhu and He, 2021). The motivation for creating this novel approach was to address memory inefficiencies in in-batch training. However, while the authors are correct in claiming that the memory inefficiencies are due to large batch sizes, they fail to address the mini-batch size which in most models is a tunable hyper-parameter that can reduce memory overhead and still offer decent performance as highlighted by the Mixed Negative Sampling approach. Moreover, the authors later acknowledge that increasing the size of the memory bank is detrimental to the stability of the item embeddings which degrades the model performance.

One may surmise that on balance, the motivations to propose the Cross-Batch Negative Sampling approach may have been unnecessary given that it is a slightly more complex model and as discussed, questionable whether the potential gains warrant the added complexity of introducing a memory bank to the model. Nevertheless, the authors were able to demonstrate that on experiments run on Youtube recommendations (Covington, Adams and Sargin, 2016), Cross-Batch Negative Sampling converged quicker with better recall and nDCG over Mixed Negative Sampling, making it a viable approach to test at scale on large corpora.

Cache-Augmented Inbatch Importance Resampling (*X*IR)

Motivated by the sampling bias within in-batch sampling strategies, Chen et al. (2022) proposed a novel Cache-Augmented Inbatch Importance Resampling (*X*IR) that resamples items from a mini-batch based on adjusted probabilities by introducing a cache that stores frequently sampled item embeddings to augment the candidate set. The inspiration for the Cache-Augmented Inbatch Importance Resampling came from the previous novel Mixed Negative Sampling approach. However, this approach increases complexity which is acknowledged in the study: the resampling of the mini-batch incurs additional memory and computational costs which is further exacerbated by the additional cost of memory for caching the items with the highest scores.

Despite the complexity to the model, when experimented on a smaller Amazon dataset, it performed better than Mixed Negative Sampling on recall and normalised discounted cumulative

gain. However, it is quite likely that the model will struggle to scale into the realms of large corpora due to the required additional computational and memory needs during training.

Batch-Mix Negative Sampling

Batch-Mix Negative Sampling is yet another model that addresses the sampling bias issue by using a statistical method to generate additional negative samples based on the the mini-batch and uses the item frequency in this virtual list to select the right negatives for training (Fan et al., 2023). This approach reduces computational complexity and memory overheads unlike previously discussed approaches of Cross-Batch Negative Sampling and Cache-Augmented Inbatch Importance Resampling. To prove this, Batch-Mix Negative sampling has a space complexity of $O(Kx|B|)$ unlike Cache-Augmented Inbatch Importance Resampling which has a space complexity of $O(|B|x|B|) + O(|C| + N)$ where B is the mini-batch, K is the negative sample and C is the cache size.

When experimented on the Amazon dataset, Batch-Mix Negative sampling had significantly better recall and normalised discounted cumulative gain over Mixed Negative Sampling. Comparing this data against the improvements that Cache-augmented Inbatch Resampling realised, Batch-Mix Negative Sampling is a more performant approach with reduced computational costs and memory overheads.

Bayesian Negative Sampling

Liu and Wang (2023) proposed Bayesian Negative Sampling to identify true negatives by designing a Bayesian classifier and creating a Bayesian optimal sampling rule to sample desired negatives, thus improving the quality of negative sampling. Experiments using this approach were run on the Movie Lens dataset with collaborative filtering and found to perform significantly better than Random Negative Sampling, Popularity-based Negative Sampling and Dynamic Negative Sampling on precision, recall and normalised discounted cumulative gain. While this specific approach was not experimented within the context of a two-tower network, given the low computational costs to create the sampled negatives, this will like scale to large corpora and two-tower models.

2.2.2 Evaluation Metrics

In Section 2.2.1, a number of metrics were mentioned in assessing the performance of the state-of-the-art methods. These include hit ratio, mean reciprocal rank (MRR), precision@k, recall@k and normalised discounted cumulative gain (nDCG). Wang (2021) defines each of these metrics and presents their formulas as mentioned below.

Hit Ratio

Hit ratio is defined as the correct answer given to a proportion of users U within a recommendation list of length L. In equation 2, U^L_{hit} is the proportion of users that have the correct answer included within the recommendation list.

$$HR = \frac{|U^L_{hit}|}{|U_{all}|} \tag{2}$$

Precision and Recall

precision@k

precision@k is defined as the proportion of relevant items in the top k recommendations from all retrieved items. Precision indicates how well the model predicted the relevant items and is formally presented as

$$precision = \frac{\{relevant\ items\} \cap \{retrieved\ items\}}{|\{retrieved\ items\}|} \tag{3}$$

recall@k

Conversely, recall@k is defined as the proportion of relevant items in the top k recommendations from all relevant items. Recall indicates how well the model did when predicting the relevant items and is formally presented as

$$recall = \frac{\{relevant\ items\} \cap \{retrieved\ items\}}{|\{relevant\ items\}|} \tag{4}$$

Mean Reciprocal Rank (MRR)

Reciprocal Rank

The reciprocal rank of a user, $RR(u)$, is defined as the total relevance score of the top L items, each weighted by its reciprocal rank and is formally presented as

$$RR(u) = \sum_{i \leq L} \frac{relevance_i}{rank_i} \tag{5}$$

Mean Reciprocal Rank (MRR) is therefore defined as the mean of all users that have received a reciprocal rank and is formally presented as

$$MRR = \frac{1}{|U_{all}|} \sum_{u=1}^{|U_{all}|} RR(u) \tag{6}$$

Normalised Discounted Cumulative Gain (nDCG)

Normalised discounted cumulative gain (nDCG) is preferred as a metric in recommendation systems as it improves upon Discounted Cumulative Gain (DCG) and Cumulative Gain (CG).

Gain

An item's **gain** is typically the relevancy score, and in the context of recommendation systems will be binary where 1 indicates the user has interacted with the item and 0 indicates the user has not interacted with the item.

Cumulative Gain

CUmulative gain is the sum of gains up to position k in the recommendation list. It does not account for the order of items in the list which is a problem for recommendation systems

since the most relevant items in a recommendation should be ranked first. Cumulative gain is formally presented as

$$CG(k) = \sum_{i=1}^{k} G_i \tag{7}$$

Discounted Cumulative Gain (DCG)
Discounted Cumulative Gain overcomes the ranking limitation of cumulative gain by dividing the gain by its rank. This guarantees that highly relevant items are placed at the top of the recommendation list. However, discounted cumulative gain is unable to compare against different top k as the score increases with k and is not indicative of the quality of the recommendation. Discounted cumulative gain is formally presented as

$$DCG(k) = \sum_{i=1}^{k} \frac{G_i}{log_2(i+1)} \tag{8}$$

Ideal Discounted Cumulative Gain
To overcome the computed score limitation of discounted cumulative gain, ideal discounted cumulative gain re-ranks the items according to their relevance in descending order up to position k. Replacing k in discounted cumulative gain with $|I(k)|$, ideal discounted cumulative gain is formally presented as

$$IDCG(k) = \sum_{i=1}^{|I(k)|} \frac{G_i}{log_2(i+1)} \tag{9}$$

Normalised Discounted Cumulative Gain (nDCG)
nDCG is therefore simply the normalisation of discounted cumulative gain over ideal discounted cumulative gain and is formally presented as

$$nDCG(k) = \frac{DCG(k)}{iDCG(k)} \tag{10}$$

2.3 Applications of Negative Sampling in Large Corpora

More recent developments in the last few years around recommendation systems have been in extreme multi-label classification where the labels in the corpus scale to millions and billions of labels. At this scale, the challenge is to remain computationally and memory efficient while ensuring a fast convergence. A good negative sampling strategy helps attain this goal (Dahiya et al., 2022; Gupta et al., 2023). In the development of negative mining-aware mini batching for extreme classification (NGAME), Dahiya et al. (2022) found that they were able to achieve a 25% - 40% faster convergence compared to other techniques by feeding the initial model with easy negatives before progressively applying harder negatives in subsequent epochs and yet still maintain low computational costs and memory efficiency.

Youtube, one of the world's largest online video content catalog, frames the problem of recommendation as an extreme multiclass classification task due to their enormous corpus. Youtube uses a method of candidate sampling to select a smaller set of negative samples from the overall corpus before adjusting these negative samples using a technique called importance weighting to ensure accuracy (Covington, Adams and Sargin, 2016). Similarly, Pinterest samples negative examples via an alternate candidate sampling technique using session co-occurrence (Liu et al., 2017) for related pins, a web-scale recommendation system that powers over 40% of user engagement on Pinterest.

Later works by Yi et al. (2019) show that they were able to reduce bias on extreme multi-label classification tasks by considering only in-batch items as negatives and *logQ* correction. They show that on live Youtube experiments, the sampling-bias-corrected model (Yi et al., 2019) was able to achieve significant performance gains over standard softmax loss computation.

2.4 Summary

This chapter initially provided a comprehensive overview of recommendation systems and their pivotal role in today's digital landscape.Recommendation systems enhance user engagement and retention by delivering personalised content, a crucial factor for long-term user satisfaction.

The focus then shifted to two-tower neural networks and the concept of negative sampling, which are central to this review. Section 2.1 detailed the architecture of the two-tower network highlighting its simplicity, scalability and flexibility. These attributes make it a versatile model for various recommendation system applications. Despite this versatility, three challenges within the architecture were also presented. Namely, the cold-start issue, selection bias and the lack of interaction between the two towers were discussed with validation from previous studies.

To address these issues, the review explored the evolution of negative sampling techniques. Initially, strategies like random negative sampling and popularity-based negative sampling were discussed. The review then emphasised the importance of hard negative sampling strategies, which have shown to improve a model's performance by focusing on more challenging negative samples. The discussion then extended to state-of-the-artmethods in negative sampling, forming the basis for the experiments in this research. Section 2.2.2 clarified the evaluation metrics used in assessing these methods, such as precision@k, recall@k and nDCG@k.

Real world implementations of negative sampling in two-tower architectures were also examined within the context of large corpora. Section 2.3 introduced extreme multi-label classification, where maintaining low computational costs and remaining memory efficient while accelerating convergence during training are paramount. The successful applications by platforms like Youtube and Pinterest were highlighted, demonstrating significant performance improvements.

In conclusion, the scale of enterprise recommendation systems has influenced the evolution of negative sampling techniques within two-tower networks. While current state-of-the-art methods have improved the accuracy of relevant item predictions, challenges such as false negatives persist (Yang et al., 2024). Future research should focus on eliminating false negatives during training without any additional computational overhead, paving the way for more robust and efficient recommendation systems.

Chapter 3

Experiments

3.1 Design

3.2 Training

3.2.1 Limitations

Macbook M1 Pro (Metal Performance Shader)

An Apple Silicon Macbook M1 Pro was initially used to train on a small dataset. Table 1 shows the hardware specifications of this macbook.

CPU Cores	**10**
Memory	**32 GB**
GPU Cores	**16**
GPU Memory	**No dedicated GPU Memory**

Table 1: Apple M1 Pro Specification used originally at the start of research

The Apple M1 offers Metal Performance Shaders (MPS) as an alternative to NVidia's CUDA. While training a model on a small dataset, the EURLex-4K was possible in about 45 minutes, training on a larger model, the LF-AmazonTitles-131K dataset would have taken days before seeing results. Additionally the batch size needed to be significantly reduced, from 3000 down to about 300 to train due to memory limitations.

PyTorch MPS Support

PyTorch's support for MPS has not reached parity with CUDA yet. In particular, autocasting is not possible as of this writing: https://github.com/pytorch/pytorch/issues/88415. Autocasting allows using float16 thus reducing the memory footprint of training a model, however on the Mac, each epoch uses float32.

3.2.2 University Cloud

Chapter 4

Results

4.1 Simple Recommendation system with Collaborative Filtering

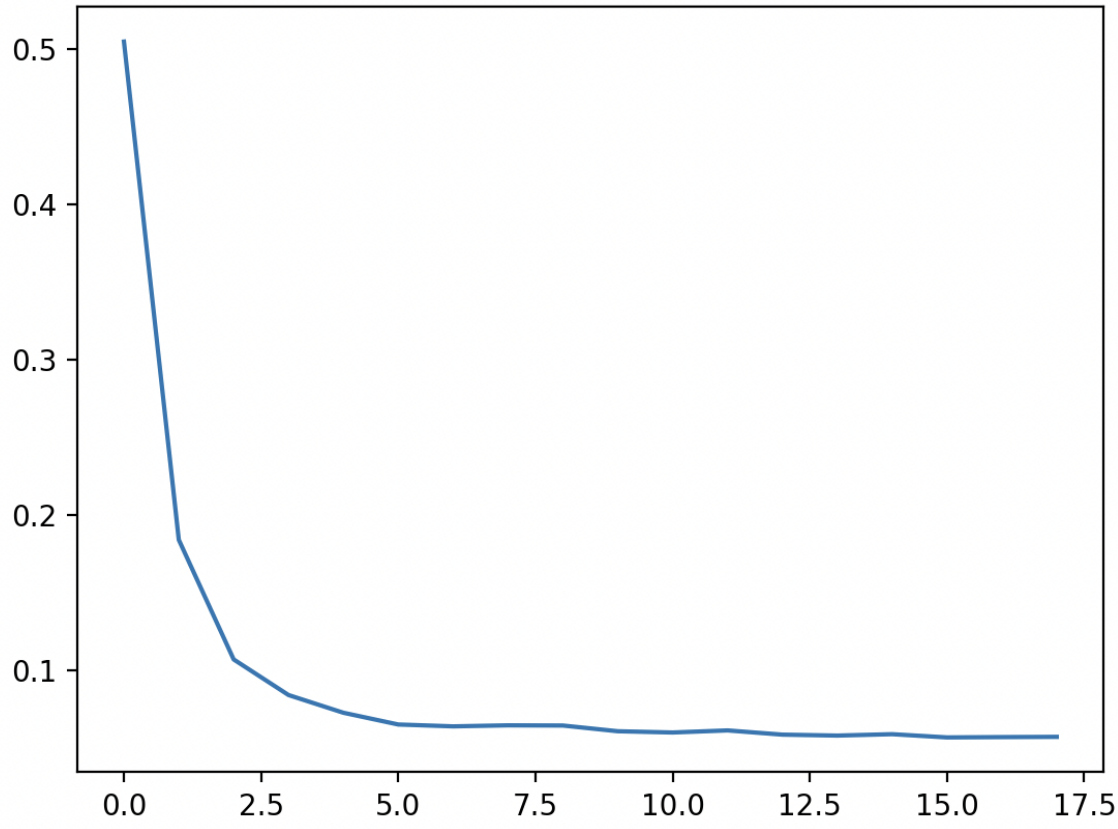

Figure 1: Training Loss over 1 Epoch

4.2 Dual Encoder models for XMC

4.2.1 Training on EURLex-4K

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
86.24	73.72	60.46	86.24	77.06	70.35	91.38	72.09	87.24	91.12

Table 1: EURLex 4K using decoupled-softmax loss

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
80.13	69.41	58.28	80.13	72.30	67.06	87.92	72.39	88.92	92.59

Table 2: EURLex 4K using softmax loss

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
84.06	72.15	60.36	84.06	75.38	69.66	90.10	73.39	87.06	90.39

Table 3: EURLex 4K using topK loss

4.2.2 Training on LF-Amazon-131K

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
42.35	28.79	20.60	42.35	44.13	46.27	50.03	56.27	65.42	68.61

Table 5: Amazon 131K using decoupled-softmax loss

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
42.15	28.54	20.47	42.15	43.84	46.0	49.85	56.17	65.49	68.83

Table 6: Amazon 131K using softmax loss

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
21.37	14.49	10.41	21.37	22.28	23.39	25.34	28.6	33.42	35.17

Table 7: Amazon 131K using decoupled softmax loss with hard negative mining

4.2.3 Training on LF-Amazon-1.3M

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
57.85	50.29	44.97	57.85	55.24	53.74	65.42	36.09	57.15	63.76

Table 8: Amazon 1.3M using decoupled softmax loss

4.2.4 Training on MovieLens-100K

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
100.0	33.33	20.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Table 9: MovieLens 100K using decoupled softmax loss

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
85.90	73.19	59.95	85.90	76.62	69.87	91.25	72.05	86.78	90.80

Table 4: EURLex 4K using decoupled-softmax loss with hard negative mining

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
0.11	0.1	0.08	0.11	0.2	0.25	0.25	0.75	4.22	7.95

Table 10: MovieLens 100K using softmax loss

P@1	P@3	P@5	nDCG@1	nDCG@3	nDCG@5	MRR@10	R@10	R@50	R@100
21.37	14.49	10.41	21.37	22.28	23.39	25.34	28.6	33.42	35.17

Table 11: Amazon 131K using decoupled softmax loss with hard negative mining

Chapter 5

Outcomes and Analysis

This is the chapter in which you review your design decisions at various levels and critique the design process.

Chapter 6

Conclusions and Future Work

This is the chapter in which you review the major achievements in the light of your original objectives, critique the process, critique your own learning and identify possible future work.

Bibliography

Aggarwal, C.C., 2016. *Recommender Systems* [Online]. Cham: Springer International Publishing. Available from: https://doi.org/10.1007/978-3-319-29659-3.

Bhatia, K., Dahiya, K., Jain, H., Kar Purushottam, Mittal Anshul, Prabhu Yashoteja and Varma, M., 2016. The extreme classification repository: Multi-label datasets and code [Online]. Available from: http://manikvarma.org/downloads/XC/XMLRepository.html.

Chen, C., Ma, W., Zhang, M., Wang, C., Liu, Y. and Ma, S., 2023. Revisiting Negative Sampling vs. Non-sampling in Implicit Recommendation. *Acm transactions on information systems* [Online], 41(1), 2, p.12. Available from: https://doi.org/10.1145/3522672/ASSET/3E3B286C-FA7A-45F7-BD37-D0C7ECC12F16/ASSETS/GRAPHIC/TOIS-2021-0138-F03.JPG.

Chen, C., Zhang, M., Zhang, Y., Liu, Y. and Ma, S., 2020. Efficient Neural Matrix Factorization without Sampling for Recommendation. *Acm transactions on information systems (tois)* [Online], 38(2), 1, p.14. Available from: https://doi.org/10.1145/3373807.

Chen, E. and Wang, B., 2024. One Backpropagation in Two Tower Recommendation Models [Online]. Available from: https://arxiv.org/abs/2403.18227v3.

Chen, J., Lian, D., Li, Y., Wang, B., Zheng, K. and Chen, E., 2022. Cache-Augmented Inbatch Importance Resampling for Training Recommender Retriever. *Advances in neural information processing systems*, 35, 12, pp.34817–34830.

Chicco, D., 2021. Siamese Neural Networks: An Overview. *Methods in molecular biology* [Online], 2190, pp.73–94. Available from: https://doi.org/10.1007/978-1-0716-0826-5{_}3/FIGURES/1.

Covington, P., Adams, J. and Sargin, E., 2016. Deep neural networks for youtube recommendations. *Recsys 2016 - proceedings of the 10th acm conference on recommender systems* [Online], 9, pp.191–198. Available from: https://doi.org/10.1145/2959100.2959190/SUPPL{_}FILE/P191.MP4.

Cui, Y., Liang, S. and Zhang, Y.Y., 2024. Multimodal representation learning for tourism recommendation with two-tower architecture. *Plos one* [Online], 19(2), 2, p.e0299370. Available from: https://doi.org/10.1371/JOURNAL.PONE.0299370.

Da, F., Kou, G. and Peng, Y., 2022. Deep learning based dual encoder retrieval model for citation recommendation. *Technological forecasting and social change* [Online], 177, 4, p.121545. Available from: https://doi.org/10.1016/J.TECHFORE.2022.121545.

Dahiya, K., Agarwal, A., Saini, D., K, G., Jiao, J., Singh, A., Agarwal, S., Kar, P. and Varma, M., 2021. SiameseXML: Siamese Networks meet Extreme Classifiers with 100M Labels [Online]. Available from: https://proceedings.mlr.press/v139/dahiya21a.html.

Dahiya, K., Gupta, N., Saini, D., Soni, A., Wang, Y., Dave, K., Jiao, J., Gururaj, K., Dey, P., Singh, A., Hada, D., Jain, V., Paliwal, B., Mittal, A., Mehta, S., Ramjee, R., Agarwal, S., Kar, P. and Varma, M., 2022. NGAME: Negative Mining-aware Mini-batching for Extreme Classification. *Wsdm 2023 - proceedings of the 16th acm international conference on web search and data mining* [Online], 7, pp.258–266. Available from: https://doi.org/10.1145/3539597.3570392.

Ding, J., Quan, Y., He, X., Li, Y. and Jin, D., 2019. Reinforced negative sampling for recommendation with exposure data. *Ijcai international joint conference on artificial intelligence* [Online], 2019-August, pp.2230–2236. Available from: https://doi.org/10.24963/IJCAI.2019/309.

Fan, Y., Chen, J., Jiang, Y., Lian, D., Guo, F. and Zheng, K., 2023. Batch-Mix Negative Sampling for Learning Recommendation Retrievers. *International conference on information and knowledge management, proceedings* [Online], 10, pp.494–503. Available from: https://doi.org/10.1145/3583780.3614789.

Felfernig, A., Polat-Erdeniz, S., Uran, C., Reiterer, S., Atas, M., Tran, T.N.T., Azzoni, P., Kiraly, C. and Dolui, K., 2019. An overview of recommender systems in the internet of things. *Journal of intelligent information systems* [Online], 52(2), 4, pp.285–309. Available from: https://doi.org/10.1007/S10844-018-0530-7/TABLES/16.

Gupta, N., Khatri, D., Rawat, A.S., Bhojanapalli, S., Jain, P. and Dhillon, I.S., 2023. Dual-Encoders for Extreme Multi-Label Classification. *Iclr 2024 camera-ready publication* [Online], 10. Available from: https://arxiv.org/abs/2310.10636v2.

Gupta, S. and Dave, M., 2020. An Overview of Recommendation System: Methods and Techniques. *Proceedings of icacm 2019* [Online], pp.231–237. Available from: https://doi.org/10.1007/978-981-15-0222-4{_}20.

Harper, F.M. and Konstan, J.A., 2015. The MovieLens Datasets. *Acm transactions on interactive intelligent systems (tiis)* [Online], 5(4), 12. Available from: https://doi.org/10.1145/2827872.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.S., 2017. Neural Collaborative Filtering. *26th international world wide web conference, www 2017* [Online], 8, pp.173–182. Available from: https://doi.org/10.1145/3038912.3052569.

Jain, H., Balasubramanian, V., Chunduri, B. and Varma, M., 2019. SlicE: Scalable linear extreme classifiers trained on 100 million labels for related searches. *Wsdm 2019 - proceedings of the 12th acm international conference on web search and data mining* [Online], 1, pp.528–536. Available from: https://doi.org/10.1145/3289600.3290979.

Jeong, H.J. and Kim, M.H., 2019. HGGC: A hybrid group recommendation model considering group cohesion. *Expert systems with applications* [Online], 136, 12, pp.73–82. Available from: https://doi.org/10.1016/J.ESWA.2019.05.054.

Jiang, T., Wang, D., Sun, L., Yang, H., Zhao, Z. and Zhuang, F., 2021. LightXML: Transformer with Dynamic Negative Sampling for High-Performance Extreme Multi-label Text Classification. *Proceedings of the aaai conference on artificial intelligence* [Online], 35(9), 5, pp.7987–7994. Available from: https://doi.org/10.1609/AAAI.V35I9.16974.

Jugovac, M. and Jannach, D., 2017. Interacting with Recommenders—Overview and Research Directions. *Acm transactions on interactive intelligent systems (tiis)* [Online], 7(3), 9. Available from: https://doi.org/10.1145/3001837.

Lee, W.M. and Cho, Y.S., 2023. A Flexible Two-Tower Model for Item Cold-Start Recommendation. *Ieee access* [Online], 11, pp.146194–146207. Available from: https://doi.org/10.1109/ACCESS.2023.3346918.

Li, R., Deng, W., Cheng, Y., Yuan, Z., Zhang, J. and Yuan, F., 2023. Exploring the Upper

Limits of Text-Based Collaborative Filtering Using Large Language Models: Discoveries and Insights [Online]. Available from: https://arxiv.org/abs/2305.11700v1.

Li, X., Chen, B., Guo, H., Li, J., Zhu, C., Long, X., Li, S., Wang, Y., Guo, W., Mao, L., Liu, J., Dong, Z. and Tang, R., 2022. IntTower: The Next Generation of Two-Tower Model for Pre-Ranking System. *International conference on information and knowledge management, proceedings* [Online], 10, pp.3292–3301. Available from: https://doi.org/10.1145/3511808.3557072.

Liu, B. and Wang, B., 2023. Bayesian Negative Sampling for Recommendation. *Proceedings - international conference on data engineering* [Online], 2023-April, pp.749–761. Available from: https://doi.org/10.1109/ICDE55515.2023.00063.

Liu, D.C., Rogers, S., Shiau, R., Kislyuk, D., Ma, K.C., Zhong, Z., Liu, J. and Jing, Y., 2017. Related Pins at Pinterest: The Evolution of a Real-World Recommender System. *26th international world wide web conference 2017, www 2017 companion* [Online], 2, pp.583–592. Available from: https://doi.org/10.1145/3041021.3054202.

Lou, J., Wen, H., Lv, F., Zhang, J., Yuan, T. and Li, Z., 2022. Re-weighting Negative Samples for Model-Agnostic Matching. *Sigir 2022 - proceedings of the 45th international acm sigir conference on research and development in information retrieval* [Online], 7, pp.1823–1827. Available from: https://doi.org/10.1145/3477495.3532053/SUPPL{_}FILE/SIGIR22-SP1206.MP4.

Melchiorre, A.B., Rekabsaz, N., Parada-Cabaleiro, E., Brandl, S., Lesota, O. and Schedl, M., 2021. Investigating gender fairness of recommendation algorithms in the music domain. *Information processing & management* [Online], 58(5), 9, p.102666. Available from: https://doi.org/10.1016/J.IPM.2021.102666.

Panda, D.K. and Ray, S., 2022. Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. *Journal of intelligent information systems* [Online], 59(2), 10, pp.341–366. Available from: https://doi.org/10.1007/S10844-022-00698-5/FIGURES/2.

Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L., 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *Proceedings of the 25th conference on uncertainty in artificial intelligence, uai 2009* [Online], 5, pp.452–461. Available from: https://arxiv.org/abs/1205.2618v1.

Schifferer, B., 2021. Using Neural Networks for Your Recommender System | NVIDIA Technical Blog [Online]. Available from: https://developer.nvidia.com/blog/using-neural-networks-for-your-recommender-system/.

Shan, H., Zhang, Q., Liu, Z., Zhang, G. and Li, C., 2023. Beyond Two-Tower: Attribute Guided Representation Learning for Candidate Retrieval. *Acm web conference 2023 - proceedings of the world wide web conference, www 2023* [Online], 4, pp.3173–3181. Available from: https://doi.org/10.1145/3543507.3583254.

Shi, W., Chen, J., Feng, F., Zhang, J., Wu, J., Gao, C. and He, X., 2023. On the Theories Behind Hard Negative Sampling for Recommendation. *Acm web conference 2023 - proceedings of the world wide web conference, www 2023* [Online], 4, pp.812–822. Available from: https://doi.org/10.1145/3543507.3583223.

Su, L., Yan, F., Zhu, J., Xiao, X., Duan, H., Zhao, Z., Dong, Z. and Tang, R., 2023. Beyond Two-Tower Matching: Learning Sparse Retrievable Cross-Interactions for Recommendation. *Sigir 2023 - proceedings of the 46th international acm sigir conference on research and development in information retrieval* [Online], 7, pp.548–557. Available from: https://doi.org/10.1145/3539618.3591643.

Wang, B., 2021. Ranking Evaluation Metrics for Recommender Systems. Available from: https://towardsdatascience.com/ranking-evaluation-metrics-for-recommender-systems-263d0a66ef54.

Wang, J. and Huang, J., 2023. Explaining Negative Sampling in Recommender Systems | BCG X [Online]. Available from: https://www.bcg.com/x/the-multiplier/explaining-negative-sampling-in-recommender-systems.

Wang, J., Zhu, J. and He, X., 2021. Cross-Batch Negative Sampling for Training Two-Tower Recommenders. *Sigir 2021 - proceedings of the 44th international acm sigir conference on research and development in information retrieval* [Online], 7, pp.1632–1636. Available from: https://doi.org/10.1145/3404835.3463032.

Yan, L., Qin, Z., Zhuang, H., Wang, X., Bendersky, M. and Najork, M., 2022. Revisiting Two-tower Models for Unbiased Learning to Rank. *Sigir 2022 - proceedings of the 45th international acm sigir conference on research and development in information retrieval* [Online], 7, pp.2410–2414. Available from: https://doi.org/10.1145/3477495.3531837/SUPPL{_}FILE/SIGIR22-SP1875.MP4.

Yang, Z., Ding, M., Huang, T., Cen, Y., Song, J., Xu, B., Dong, Y. and Tang, J., 2024. Does Negative Sampling Matter? A Review with Insights into its Theory and Applications. *Ieee transactions on pattern analysis and machine intelligence* [Online]. Available from: https://doi.org/10.1109/TPAMI.2024.3371473.

Yang Google, J., Yi Google, X., Zhiyuan Cheng Google, D., Hong Google, L., Li Google, Y., Xiaoming Wang Google, S., Xu Google, T., Chi Google, E.H., Yang, J., Yi, X., Zhiyuan Cheng, D., Hong, L., Li, Y., Xiaoming Wang, S., Xu, T. and Chi, E.H., 2020. Mixed Negative Sampling for Learning Two-tower Neural Networks in Recommendations. *The web conference 2020 - companion of the world wide web conference, www 2020* [Online], 7, 4, pp.441–447. Available from: https://doi.org/10.1145/3366424.3386195.

Yi, X., Yang, J., Hong, L., Cheng, D.Z., Heldt, L., Kumthekar, A., Zhao, Z., Wei, L. and Chi, E., 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. *Recsys 2019 - 13th acm conference on recommender systems* [Online], 9, pp.269–277. Available from: https://doi.org/10.1145/3298689.3346996.

Yu, Y., Wang, W., Feng, Z. and Xue, D., 2021. A Dual Augmented Two-tower Model for Online Large-scale Recommendation. *Dlp-kdd 2021* [Online]. Available from: https://doi.org/10.1145/1122445.1122456.

Zhao, Y., Chen, R., Lai, R., Han, Q., Song, H. and Chen, L., 2023. Augmented Negative Sampling for Collaborative Filtering. *Proceedings of the 17th acm conference on recommender systems, recsys 2023* [Online], 11(23), 9, pp.256–266. Available from: https://doi.org/10.1145/3604915.3608811.

Appendix A

Code

A.1 File: yourCodeFile.java

```
// This is an example java code file, just for illustration
    purposes
public static void main() {
  System.out.print ("Hello World");
}
```