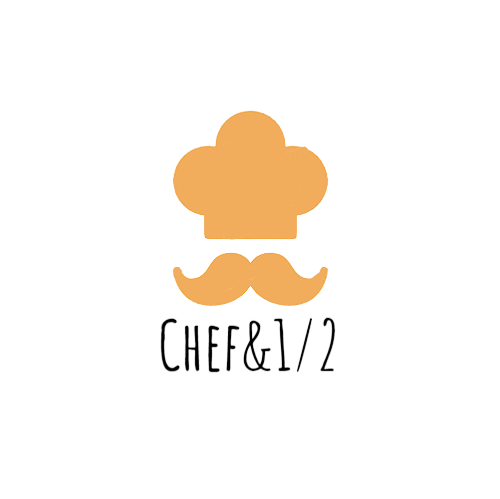


Cairo University

Faculty of Engineering

Department of Computer Engineering

**Chef w Nos**

A Graduation Project Report Submitted

to

Faculty of Engineering, Cairo University

in Partial Fulfillment of the requirements of the degree

of

Bachelor of Science in Computer Engineering.

**Presented by**

Karim Amr Mohamed Amr

Abdallah Ahmed Omar Magdy

**Supervised by**

Dr.Magda Fayek

12-06-2023

All rights reserved. This report may not be reproduced in whole or in part, by photocopying or other means, without the permission of the authors/department.

Abstract

With the fast-paced lifestyle that we are living, eating has become an afterthought, people want to eat to get through the day and don’t want to spend time thinking and planning what to eat, which makes cooking even more difficult to do daily.

Figuring out what to eat every day is needless time wasting whether it’s just you eating alone or your whole family negotiating and discussing what to eat just to settle for an old recipe.

It leads to cooking the same recipes in a cycle which heavily limits recipe discovery and can turn cooking into more of a tedious chore.

After all the thinking and planning when you finally settle on a recipe it’s hard to follow the video’s steps and you end up pausing and rewinding which makes you wonder why I cooked in the first place.

To make the cooking process easier, faster, and more convenient the user should have fast access to recipes that he’s likely to make and that suit his inventory of ingredients, and be able to ask for certain recipes, ingredients, or cravings.

The user should also have a way to conveniently navigate the steps of the recipe that he chose to make.

We propose a system that accompanies the user and helps make cooking a simpler task.

The system should be able to chat with the user to understand their craving and can adjust on the fly to the user’s requests. The user’s request is then sent to a recommendation system that recommends recipes based on user’s request, favorite recipes, and past recipes, this process can go back and forth until the user is satisfied and chooses a recipe.

The recipe is then loaded and presented to the user as a collection of steps he can move through freely.

The system is presented in an application form where users can chat with their chef and can ask for similar recipes, recipes with certain ingredients and characteristics.

The application also provides relevant recommendations based on user’s likes and history.

An inventory system is also added to monitor ingredients and give users insights on their shopping and diet.

Recipe understanding is also presented to segment recipes video to logical steps to allow the user to freely navigate the recipe’s steps.

الملخص

مع نمط الحياة السريع الذي نعيشه، أصبح الأكل مجرد فكرة ثانوية، حيث يرغب الناس في تناول الطعام للمضي قدمًا في يومهم ولا يرغبون في قضاء الوقت في التفكير والتخطيط لما يجب تناوله. وهذا يجعل عملية الطبخ أكثر صعوبة في القيام بها يوميًا.

قرار ما يجب تناوله كل يوم يستهلك الكثير من الوقت عن طريق التفاوض ومناقشة العائلة بشأن ما يجب تناوله فقط للوصول إلى وصفة قديمة. هذا يؤدي إلى تكرار نفس الوصفات بشكل متكرر، مما يقيد اكتشاف وصفات جديدة ويجعل عملية الطبخ مهمة مملة.

بعد كل التفكير والتخطيط عندما تستقر على وصفة معينة، يكون من الصعب اتباع خطوات الفيديو وتجد نفسك توقف وإعادة المشاهدة، مما يجعلك تتساءل لماذا طبخت في المقام الأول.

لجعل عملية الطبخ أسهل وأسرع وأكثر ملائمة، يجب أن يتمكن المستخدم من الوصول السريع إلى وصفات يحتمل أن يقوم بتحضيرها وتتناسب مع مخزونه من المكونات، وأن يكون بإمكانه طلب وصفات معينة أو مكونات أو رغبات.

يجب أيضًا أن يكون للمستخدم وسيلة للتنقل بسهولة في خطوات الوصفة التي اختارها.

نقترح نظامًا يرافق المستخدم ويساعده على جعل الطبخ مهمة أبسط.

يجب أن يكون النظام قادرًا على محادثة المستخدم لفهم رغباته ويمكنه التكيف مع طلبات المستخدم. يتم إرسال طلب المستخدم إلى نظام توصية يوصي بوصفات استنادًا إلى طلب المستخدم والوصفات المفضلة والوصفات السابقة للمستخدم، ويمكن أن يتم هذا العمل ذهابًا وإيابًا حتى يكون المستخدم راضيًا ويختار وصفة.

ثم يتم تحميل الوصفة وتقديمها للمستخدم على هيئة مجموعة من الخطوات التي يمكنه التنقل فيها بحرية.

يتم تقديم النظام في شكل تطبيق حيث يمكن للمستخدمين التحدث مع طاهيهم وطلب وصفات مماثلة أو وصفات تحتوي على مكونات وخصائص معينة.

يوفر التطبيق أيضًا توصيات ذات صلة استنادًا إلى تفضيلات المستخدم وتاريخه.

يتم أيضًا إضافة نظام للجرد لمراقبة المكونات وتزويد المستخدمين بإرشادات حول التسوق والنظام الغذائي.

كما يتم تقديم فهم الوصفة لتقسيم فيديو الوصفة إلى خطوات منطقية للسماح للمستخدم بالتنقل بحرية في خطوات الوصفة.

ACKNOWLEDGMENT

First, we would like to thank God for helping us through this long and challenging journey. We would also like to thank our supervisors Dr. Magda Fayek for her support, guidance, and advice. During this journey we got a lot of support from our family and friends for which we are very thankful.

Table of Contents

[Abstract ii](#_Toc136104218)

[الملخص iii](#_Toc136104219)

[ACKNOWLEDGMENT iv](#_Toc136104220)

[Table of Contents v](#_Toc136104221)

[List of Figures vii](#_Toc136104222)

[List of Tables ix](#_Toc136104223)

[List of Abbreviation xi](#_Toc136104224)

[List of Symbols xiii](#_Toc136104225)

[Contacts xiv](#_Toc136104226)

[Chapter 1: Introduction 1](#_Toc136104227)

[1.1. Motivation and Justification 2](#_Toc136104228)

[1.2. The Essential Question 2](#_Toc136104229)

[1.3. Project Objectives and Problem Definition 2](#_Toc136104230)

[1.4. Project Outcomes 3](#_Toc136104231)

[1.5. Document Organization 3](#_Toc136104232)

[Chapter 2: Market Feasibility Study 4](#_Toc136104233)

[2.1. Targeted Customers 4](#_Toc136104234)

[2.2. Market Survey 4](#_Toc136104235)

[2.2.1. BASYL[3] 4](#_Toc136104236)

[2.2.2. Tasty[4] 6](#_Toc136104237)

[Chapter 3: Literature Survey 8](#_Toc136104238)

List of Figures

Figure 1.1: Same caption as in the text ................................................................................. page

Figure 2.1: Same caption as in the text ................................................................................. page

[Figure ‎2‑1 BASYL Prompt 4](#_Toc136100365)

[Figure ‎2‑2 BASYL output 5](#_Toc136100366)

List of Tables

Table 1.1: Same caption as in the text .................................................................................. page

Table 2.1: Same caption as in the text .................................................................................. page

List of Abbreviation

[The abbreviations should be put in an alphabetical order]

AI Artificial Intelligence

EA Evolutionary Algorithms

GA Genetic Algorithms

SA Simulated Annealing

VLSI Very Large Scale Integration

List of Symbols

[The symbols should be put in an alphabetical order. Greek symbols come first, followed by English symbols]

σ Noise standard deviation

B Buffer size

fop Operating frequency

Contacts

**Team Members**

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Phone Number** |
| Name of Student 1 | [abc1@email.com](mailto:abc1@email.com) | +2 01xxxxxxxxx |
| Name of Student 2 | [abc2@email.com](mailto:abc2@email.com) | +2 01xxxxxxxxx |
| Name of Student 3 | [abc3@email.com](mailto:abc3@email.com) | +2 01xxxxxxxxx |
| Name of Student 4 | [abc4@email.com](mailto:abc4@email.com) | +2 01xxxxxxxxx |

**Supervisor**

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Number** |
| Name of Supervisor | [abc5@email.com](mailto:abc5@email.com) | +2 01xxxxxxxxx |

This page is left intentionally empty

# Introduction

The foods you choose to eat can have a direct impact on your ability to enjoy life to fullest. Perhaps the most obvious positive effect of food is the pleasurable feeling you get from eating a good-tasting meal, but to get there we have to go through a long process from first deciding what to cook until we sit down with our loved ones to eat what you have made.

Making food is no easy task it requires planning, knowledge and skill to cook something great we want to cut out the most boring parts of the experience so you can focus on the actual cooking and leave the rest out of your mind.

We all take part in cooking some way or another, whether we are the ones doing the actual cooking or being part of the discussion on what to eat today, a question that each one of us in exposed to daily and takes way more time than it is needed.

A smart cooking assistant would take care of these tedious details and would encourage people to try more recipes and discover whole new cooking styles, and would pull people back into cooking.

Having relevant recommendations in front of you and being able to ask for specific ingredients/characteristics and having all that within the items in your kitchen would greatly improve the cooking process.

We set out to simplify the planning and knowledge needed to start cooking by providing the users with tailored recipe recommendations based on their history, preferences and specific requests, while making following a recipe’s video more manageable while cooking.

The system proposed can interact with the users to get their specific requests if any using a chatbot that answers that understand requests and queries a recommendation system.

After choosing the right recipe the recipe’s video is analyzed and divided to steps to allow for smoother navigation.

## Motivation and Justification

Cooking is a not an easy task by any means due to the amount of planning, knowledge and skills needed to pull off a successful recipe and we are living in a fast-paced environment where food, groceries and taxies are being ordered by a push of a button, modernizing the cooking process would take out the boring parts leaving you with the pleasure of cooking a dish and the satisfaction that follows.

During late 2020 and early 2021 one of the common trends across the world was cooking at home[1] and that coincides with COVID-19 pandemic, people were at home and used this as an opportunity to level up their cooking skills.

Moreover, people between the age of 25-34 years old cook with either their smartphones or tablets[2] and the smartphone is becoming the ultimate sous-chef for millennials as they prefer to experience the whole culinary process from start to finish.

So, it’s apparent that people are already on their phones to look for the next recipes to make and the proposed application would fit perfectly into this category of helping people focus on cooking and exploring.

## The Essential Question

We all end up eating at the end of the day whether by cooking or ordering food but the question that we repeat to ourselves every day is what to eat today?

And eventually the main problem we are tackling, we want to eliminate this question and keep it to a minimum and we believe as engineers we should always look to make people’s life easier and more convenient by automating tedious tasks using technology.

And this mission perfectly aligns with the problem at hand it takes such an important aspect of our lives and simplifies it and make it more modern.

## Project Objectives and Problem Definition

We set out to simplify the planning and knowledge needed to start cooking so our users can focus on the cooking process itself.

The system should take care of the decision-making process as much as possible so the user can quickly start making the recipe, it also needs to react to feedback from user and receive specific requests to be as accurate as possible.

Keeping in mind the changing taste of the users the system should be able to adapt overtime to the users’ taste.

It’s important also that the recipes presented to the user are feasible, meaning that the ingredients are available in the kitchen.

In line with simplifying the knowledge needed to cook we’ll use videos that show step by step the recipe and the user can navigate easily through the video whilst following it.

Chef W Nos’s main focus is to help you decide your recipe for the day, we assume that the chatbot will only be used to ask for recipes and get positive or negative feedback not for general chatting or explaining steps of the recipe.

## Project Outcomes

The outcome of this project is a software application that accompanies the user on their android phone with a simple graphical user interface.

The user can see right away his recommended recipes that he can browse through, view their steps and attached videos.

If the user has a specific craving, he can then use the chatbot to ask for recommendation based on that craving, after choosing the recipe he’ll be directed to the recipe’s steps and segmented video that he can then navigate through.

## Document Organization

In this section, you have to give the organization of the report and a quick description of the following chapters.

# Market Feasibility Study

In this chapter we’ll discuss the market feasibility study, going into details into the targeted customers, market survey and talk about the business case and financial analysis.

Cooking assistants are pretty common in the market, when we first started to develop this idea, we looked at what other applications offered, gaining valuable insights that helped us come up with the set of features present in our applications.

## 2.1. Targeted Customers

As we discussed in the last chapter more people now are using their smartphones to help them in cooking, research from McGarry Bowen and Kraft Foods, the found that 59% of 25–34-year-olds cook with their smartphones[2].

So young people that are already using their smartphones for everything are one of our targeted customers.

Another group that we are very interested in is household parents, families tend to have the most discussion about what to eat today and with varying tastes and ingredients they would be benefit greatly from the application’s features.

## 2.2. Market Survey

In this section we’ll explore similar applications that provide cooking assistance to the user.

### 2.2.1. BASYL[3]

Basyl is an AI cooking assistant that generates recipes based on ingredients entered and can save generated recipes in a cookbook.

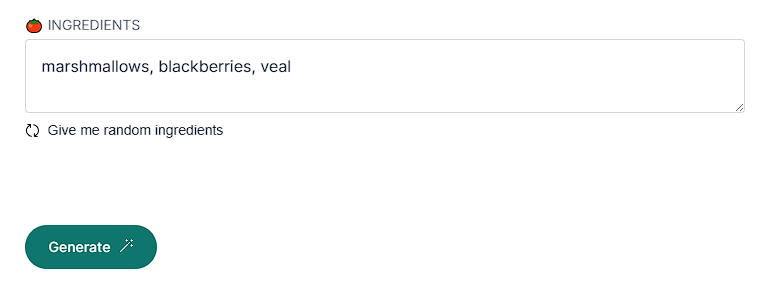


Figure ‑ BASYL Prompt

****

Figure ‑ BASYL output

**Features:**

* Generate delicious custom recipes based on your preferences.
* Save your favorite recipes to your personalized cookbook.

**Drawbacks:**

* No recipe recommendations.
* No inventory system.

### 2.2.2. Tasty[4]

A project about recommending recipes and ingredients based on the needs of the user.

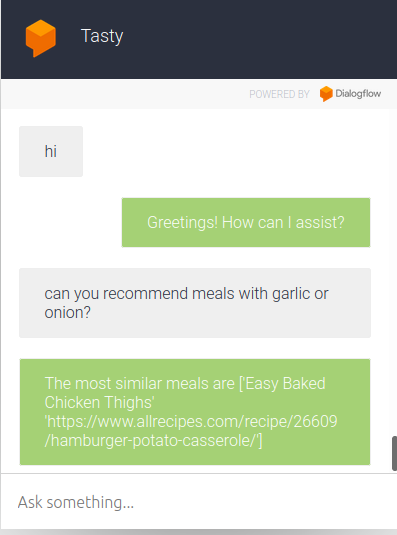
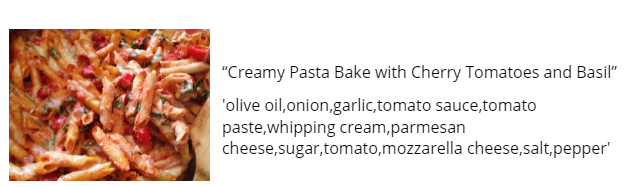
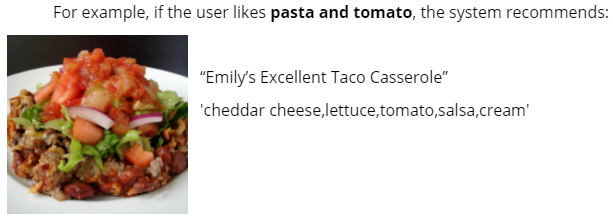


Figure ‑ Tasty's recipe recommendations

Figure ‑ Tasty chat interface

**Features:**

* Chatbot to interact with user.
* Get recipe recommendations with specified ingredients.
* Get similar ingredients and substitutions.

**Drawbacks:**

* No inventory system
* Doesn’t keep history of recipes.
* Redirects to recipe in text form.

## 2.3. Business Case and Financial Analysis

### 2.3.1. Competitive Analysis

According to our market research we found that there is no obvious competitor, as there is no application provides all our features at the same time. We found many applications each one of them provides part of our application’s features. Also, there are some features in our application we didn’t find any application provides them.

The closest application to our application is “Tasty”, as it has a recommendation system, step by step video analysis and “what’s in your kitchen feature”. But on the other hand, there is no chatbot, also there are some differences between the recommendation system and our one.

### 2.3.2. Business Case

### 2.3.3. Financial Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Measure | Year 1 | Year 2 | Year 3 | Year 4 |
| Application host & cloud server | $500 | $1,000 | $1,500 | $2,000 |
| Salaries | $6,000 | $8,000 | $10,000 | $12,000 |
| Office rent & bills | $2,500 | $2,500 | $4,300 | $4,300 |
| Advertising | $250 | $250 | $250 | $250 |
|  |  |  |  |  |
| Total cash out | $9,250 | $11,750 | $16,050 | $18,550 |
|  |  |  |  |  |
| Subscriptions profit | $1,000 | $3,000 | $8,000 | $13,000 |
| Application ads profit | $1,000 | $2,000 | $4,000 | $6,000 |
|  |  |  |  |  |
| Total cash in | $2,000 | $5,000 | $12,000 | $19,000 |
|  |  |  |  |  |
| Revenue | -$7,250 | -$6,750 | -$4,050 | $450 |

# Literature Survey

In this chapter we will introduce the topics needed to understand the project and we’ll also discuss the various papers, studies and methods explored while developing our system and speak about the challenges and difficulties faced along the way.

Going through these topics we’ll discover the importance of machine learning and AI in advancing multiple fields like computer vision, natural language processing and recommendations systems.

## 3.1. Recommender systems[5]

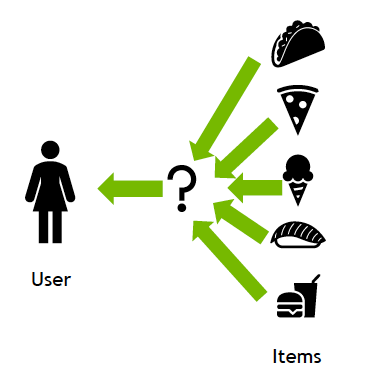
A recommendation system (or recommender system) is a class of machine learning that uses data to help predict, narrow down, and find what people are looking for among an exponentially growing number of options.[6]

Figure ‑ Recommender systems

The recommendations produced can be based on various criteria, including past purchases, search history, demographic information, and other factors.

Recommender systems have gained popularity among content and product providers due to their ability to analyze user interactions, such as impressions, clicks, likes, and purchases, in order to understand individual preferences and characteristics. By leveraging this data, these systems can make highly personalized predictions about consumer interests and desires. As a result, they can effectively steer consumers towards a wide range of products and services, including books, videos, health classes, and clothing.

### 3.1.1 Collaborative filtering[5]

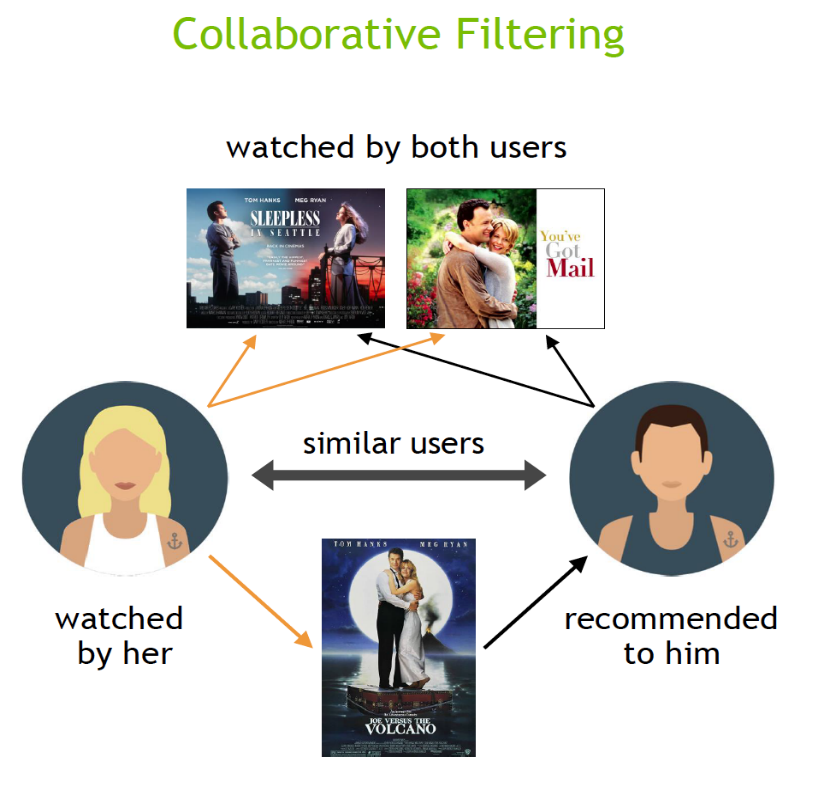
The collaborative filtering approach in recommender systems utilizes preference information from multiple users to recommend items. By analyzing the similarity of user preference behavior and considering previous interactions between users and items, these algorithms learn to predict future interactions. These systems create a model based on a user's past behavior, such as their previous purchases or ratings given to items, as well as similar decisions made by other users. The underlying concept is that if multiple users have made similar decisions and purchases in the past, such as choosing the same movie, there is a high likelihood that they will agree on future selections. For instance, if a collaborative filtering recommender system recognizes that you and another user have similar tastes in movies, it may suggest a movie to you that it knows the other user already enjoys.

Figure ‑ Collaborative filtering

### 3.1.2 Content-based filtering [5]

The content filtering approach in recommender systems relies on the attributes or features of an item to recommend other items that are similar to the user's preferences. This approach is based on analyzing the similarity between the attributes of items and the user's preferences. By considering information about a user and the items they have interacted with, such as a user's age, the cuisine category of a restaurant, or the average review for a movie, the recommender system models the likelihood of a new interaction.

For instance, if a content filtering recommender system observes that you enjoyed movies like "You've Got Mail" and "Sleepless in Seattle," it may recommend another movie to you with similar genres and/or cast, such as "Joe Versus the Volcano." The system takes into account the shared attributes or features of these movies, such as their genres or actors, to suggest other items that align with your preferences.

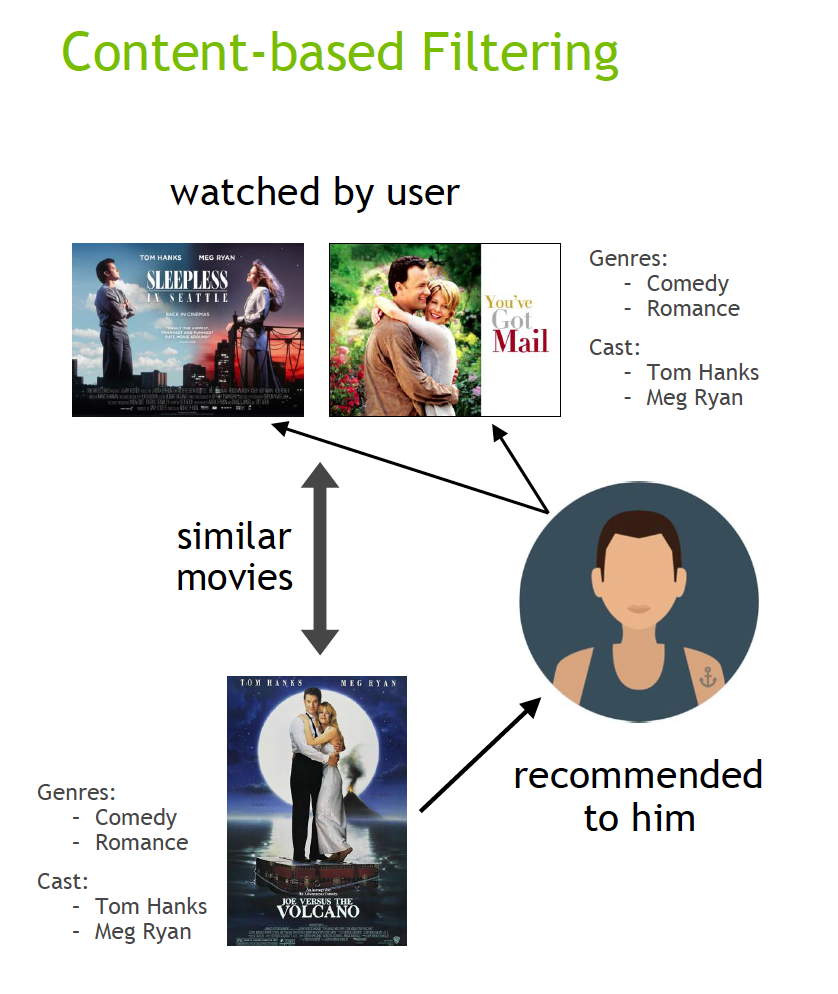


Figure ‑ Content-based

### 3.1.3 Context filtering[5]

In order to improve recommendations, Netflix has explored incorporating users' contextual information into the recommendation process. During a presentation at NVIDIA GTC, Netflix discussed their approach of framing recommendations as contextual sequence predictions. This method utilizes a sequence of contextual user actions, along with the current context, to estimate the likelihood of the next action.

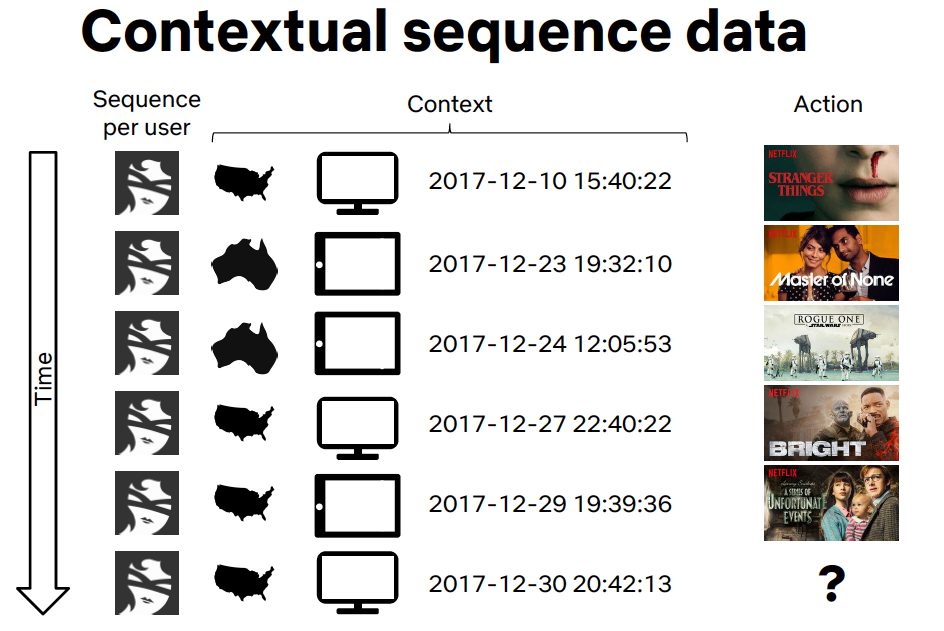
For instance, in the case of Netflix, they considered a sequence of contextual information for each user, including their country, device, date, and time when they watched a movie. By training a model using this sequence, Netflix aimed to predict what the user is likely to watch next. This approach takes into account the specific context of each user, allowing for more personalized and accurate recommendations based on their past viewing behaviors and contextual factors.

Figure ‑ Contextual Filtering

## 3.2 Dynamic Graph Neural Networks

## 3.3. Natural language processing

Historically, computers have faced challenges in truly understanding human language. While they can collect, store, and process text inputs, they often lack the fundamental context and comprehension required for effective language understanding.

To address this limitation, the field of Natural Language Processing (NLP) emerged. NLP is a branch of artificial intelligence that focuses on equipping computers with the ability to read, analyze, interpret, and derive meaning from text and spoken words. It encompasses various techniques and methodologies that combine elements of linguistics, statistics, and Machine Learning to enable computers to better understand and work with human language.

By leveraging NLP, computers can go beyond basic text processing and start to understand the intricacies of language, such as syntax, semantics, and context. This allows for a range of applications, including language translation, sentiment analysis, question-answering systems, chatbots, and more. NLP plays a crucial role in enabling computers to "understand" and interact with human language more effectively.

NLP went through many advances that enabled greater efficiency and accuracy due to the introduction of new machine learning techniques, so we are only concerned with the latest advancements.

### 3.3.1 Recurrent neural network

Recurrent neural networks (RNN) differ from regular neural network as the input to a RNN is a single word instead of a whole sentence, meaning the network can handle varying lengths of sentences.

RNNs also share feature learned across different positions of text it does so by treating each word of a sentence as a sperate input that happens at time , the input at time and the activation value at .

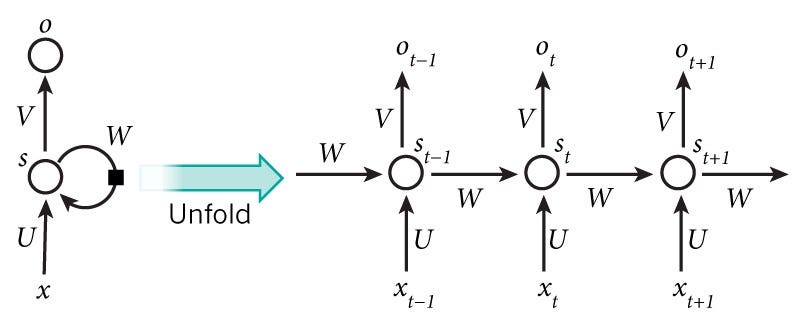


Figure ‑

### 3.3.2 Gated Recurrent Unit (GRU)

GRU (Gated Recurrent Unit) is a modified version of the basic recurrent unit used in neural networks. It addresses two important challenges: capturing long-range dependencies and overcoming the vanishing gradient problem.

The GRU introduces an update gate and an additional unit with tanh activation to improve its ability to capture long-range dependencies. This gate, along with the tanh unit, facilitates effective memory cell updating and helps overcome the vanishing gradient problem, leading to more robust and efficient training of the neural network.

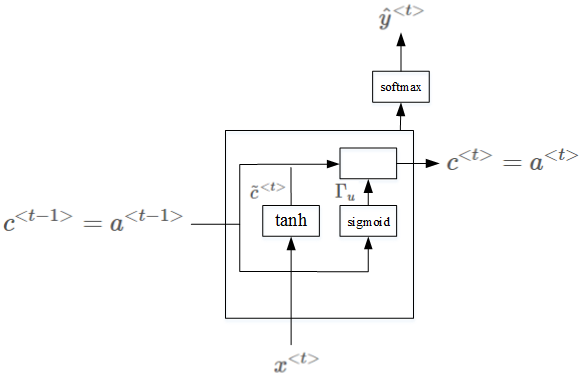


Figure ‑

### 3.3.3 LSTM

LSTM (Long Short-Term Memory) is a type of recurrent neural network architecture that is capable of addressing the challenges of capturing long-range dependencies and solving the vanishing gradient problem. It achieves this by using two gates: an update gate and a forget gate. The update gate knows how much of the new information should be incorporated into the memory cell, while the forget gate controls the extent to which the old information should be forgotten. This flexibility allows LSTM to effectively retain relevant information and discard irrelevant or outdated information, making it suitable for capturing long-term dependencies in sequential data.

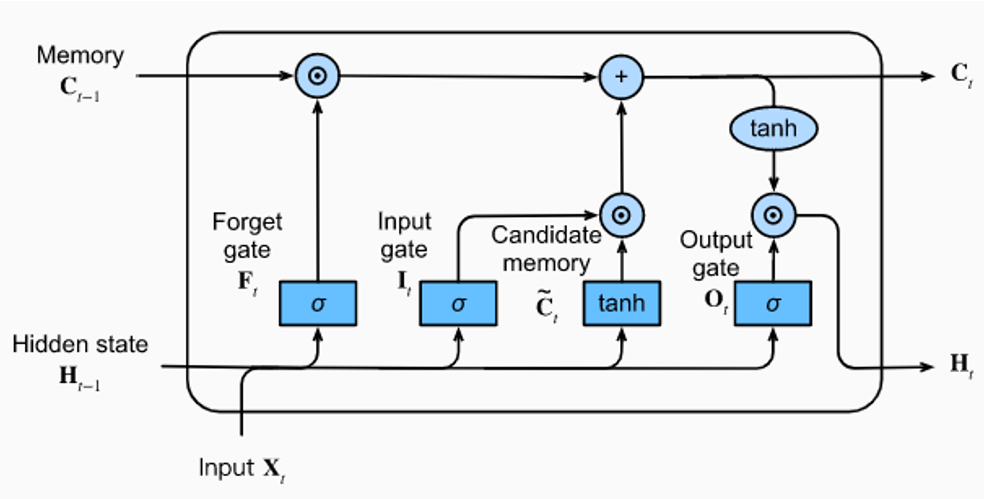


Figure ‑

## 3.4 Transformers (Attention Is All You Need)[7]

Figure ‑

Transformers is a type of neural network architecture that revolutionized natural language processing tasks. The concept of transformers was introduced in the paper "Attention Is All You Need" by Vaswani et al. in 2017.

Traditional sequence-based models, such as recurrent neural networks, suffer from limitations like sequential computation and some obstacles in capturing long-range dependencies. Transformers focused on addressing these issues by making a mechanism called self-attention.

In a transformer, the input sequence is first embedded into vectors known as embeddings. These embeddings are then passed through multiple layers of self-attention and feed-forward neural networks which they are processed in it. Self-attention allows the model to weigh the importance of different words in a sequence when generating representations, enabling it to focus on the most relevant words for each task.

The self-attention mechanism calculates attention scores between all pairs of words in a sequence. It determines how much each word is related to other words in the sequence during processing. The attention scores are computed based on the similarity of the embeddings and can be thought of as a measure of the importance of one word to another. After the self-attention step, the transformer applies feed-forward neural networks to each word's representation individually. These networks learn to transform and refine the representations. This process is repeated multiple times through stacked layers, allowing the model to figure the complex patterns and dependencies in the data.

One crucial aspect of transformers is the idea of positional encoding. Since transformers lack the sequential nature of recurrent neural networks, positional encoding is introduced to provide information about the order of words in the input sequence. This allows the model to differentiate between words based on their relative positions.

The "Attention Is All You Need" paper introduced the transformer architecture and demonstrated its effectiveness on various natural language processing tasks, such as machine translation. It showcased the superiority of transformers in capturing long-range dependencies, scalability to larger datasets, and parallelization capabilities compared to traditional recurrent neural networks.

Overall, transformers have become a fundamental architecture in the field of natural language processing, enabling breakthroughs in tasks such as machine translation, sentiment analysis, text summarization, and more. The self-attention mechanism and stacked layers of feed-forward networks in transformers have proven to be highly effective in capturing complex patterns and dependencies in sequential data.

## 3.5 (BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding)[8]

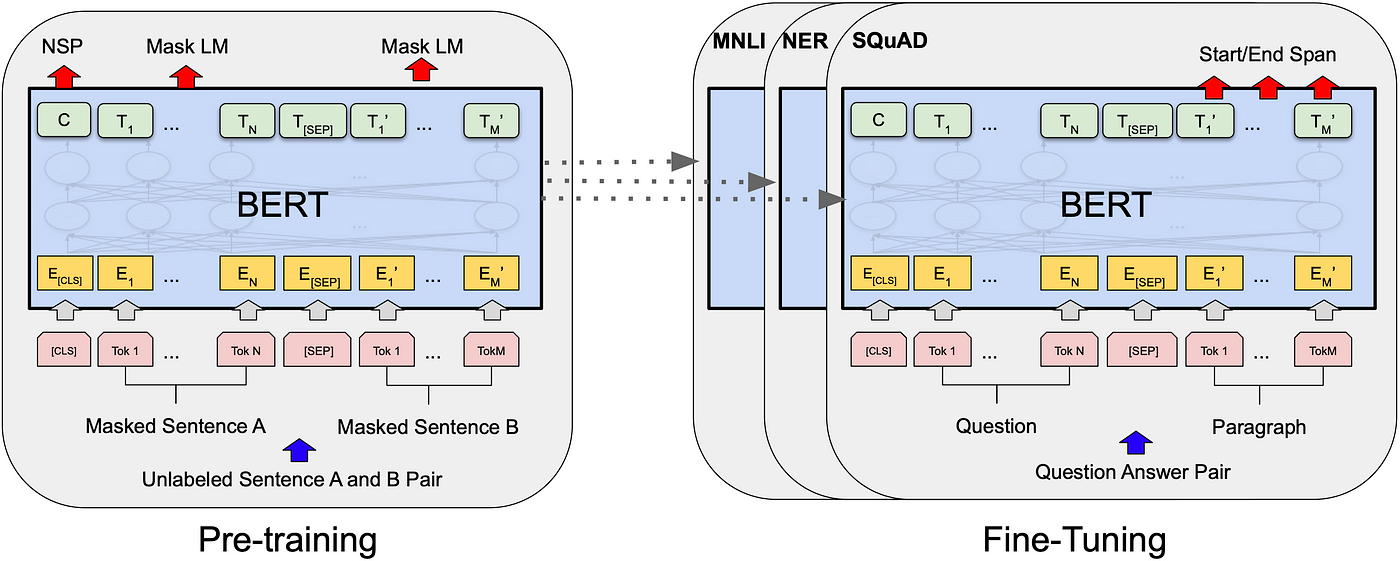


Figure ‑

BERT (Bidirectional Encoder Representations from Transformers) is a popular language model introduced in the paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" by Devlin et al. in 2018. It is a groundbreaking model that has significantly advanced natural language processing tasks.

Unlike traditional language models that process text in a sequential manner, BERT utilizes a transformer architecture and employs a pre-training and fine-tuning approach. BERT is "bidirectional" because it considers both the left and right contexts of each word during pre-training.

During pre-training, BERT is trained on large amounts of unlabeled text from various sources, such as books and websites. It learns to predict missing words in sentences by using the surrounding context. This process helps BERT develop a deep understanding of language and its nuances.

BERT's architecture consists of multiple transformer layers. Each layer includes self-attention mechanisms and feed-forward neural networks. The self-attention mechanism allows BERT to weigh the importance of different words in a sentence when generating contextualized word representations.

After pre-training, BERT is fine-tuned on specific downstream tasks, such as text classification, named entity recognition, question answering, and sentiment analysis. During fine-tuning, BERT is trained on labeled task-specific data, adjusting its parameters to make predictions for the target task.

One of the key features of BERT is its ability to capture contextual information effectively. It learns contextualized word representations that take into account the surrounding words in a sentence, which helps in understanding complex sentence structures and resolving word ambiguities.

BERT also introduces the concept of "masked language modeling" during pre-training. It randomly masks out some of the words in the input sentence and trains the model to predict those masked words based on the remaining context. This technique helps BERT to handle bidirectionality and learn deeper contextual relationships.

The impact of BERT has been significant in natural language processing, as it has achieved state-of-the-art results on various benchmark datasets. BERT's ability to capture contextual information and its transfer learning capabilities have made it a versatile and widely adopted model for various NLP tasks.

In summary, BERT is a language model that utilizes transformer architecture, pre-training, and fine-tuning to achieve remarkable results on natural language processing tasks. It learns contextualized word representations by considering bidirectional context and leverages large-scale pre-training data to develop a deep understanding of language.

## 3.6. Personalized Embedding-based e-Commerce Recommendations at eBay[9]

Items recommendations is an essential component of e-commerce marketplaces and given the scale and extreme sparsity of user-item matrix, meaning large number of new items being added and users don’t get chance to review them, the traditional collaborative filtering methods produce poor results.

Consequently, implicit user feedback such as clicks and purchases are also sparse and the traditional methods don’t fully capitalize on their significance.

In this paper an approach for generating personalized item recommendations is offered, by leaning to embed items and users in the same vector space to help combat the huge sparsity in the user-item matrix.

Item and user embeddings are computed using content features and multi-modal onsite user activity.

The model proposed for personalized recommendations is based on training a two-tower deep leaning model to generate user and item embeddings at the same time.

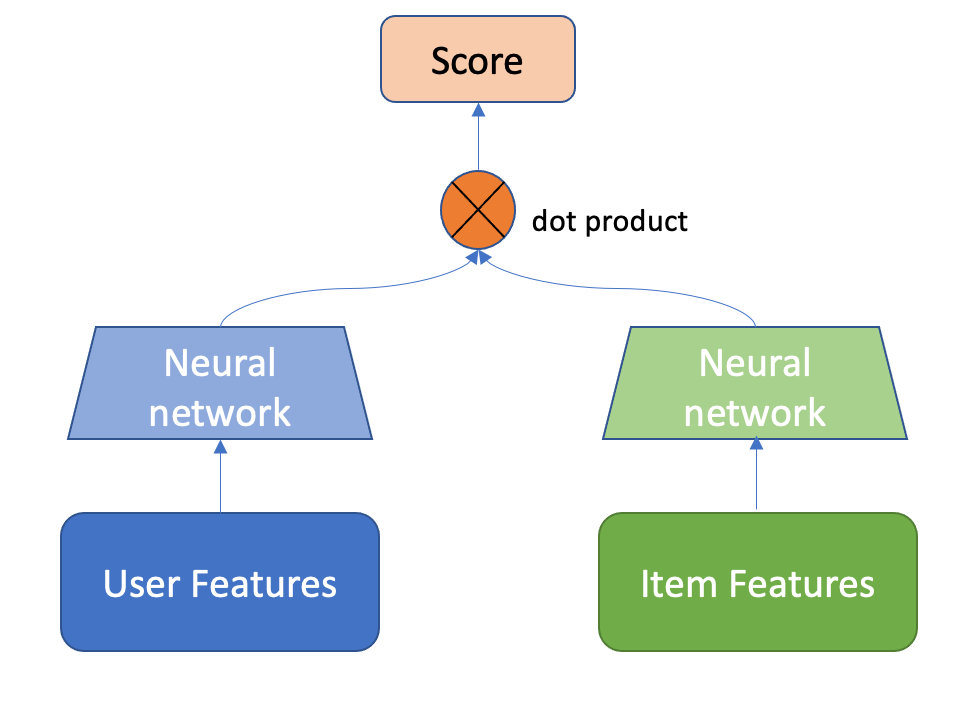


Figure ‑ Typical Two Tower model

The two-tower model is widely used for generating personal recommendations, it uses information for both the users and items together, the output of each neural network is a vector that represents user and item respectively then dot product is calculated and used as the rating of the item given by the user.

A loss function is then used to minimize the error between the actual rating and the calculated rating.

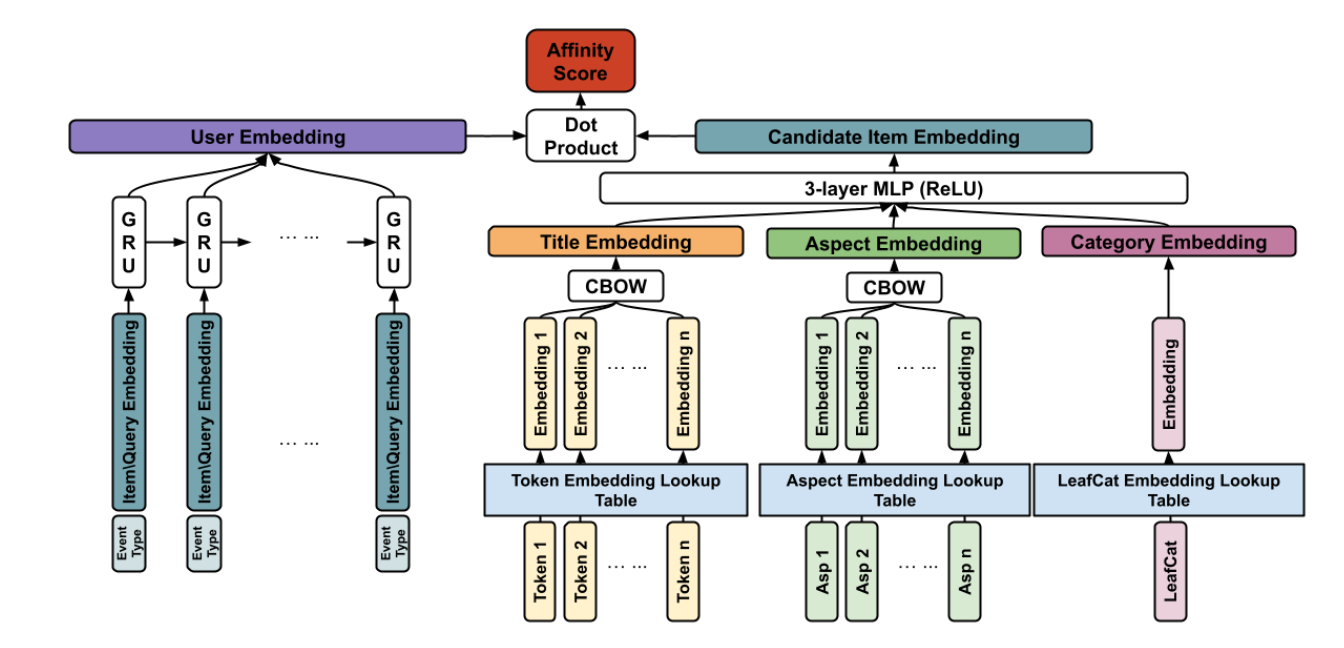


Figure ‑ eBay recommendation model

First, we’ll discuss the items network, an item at eBay corresponds to a listing, the item is represented as its content-based features such as its title, category and aspects.

Since title and aspects are text data, they are tokenized then converted to feature vectors using Continuous-Bag-Of-Words approach.

A CBOW vector is produced for each item feature then all vectors are concatenated and passed through a 3 layers neural network with ReLU activation to produce the item final embedding.

Secondly the user network, eBay considers some actions as valuable signals for the generation of recommendations such as viewing items, making a search query, adding an item to their shopping cart or their Wishlist, these actions are represented using a recurrent neural network that has access to the sequence of the events.

Each event has a GRU cell that produces output according to the following:

Where is the user event and its vector representation in .

The last part of the model is the Affinity function 𝛾 (v𝑖, u), the affinity is calculated between user U and item and it is constructed by the dot product between the user and item embeddings.

Both embeddings are normalized to have unit length and their dot product is constrained to be between -1 and 1 to distinguish between positive and negative items.

During the prediction stage, given the user embedding and a pool of candidate item embeddings the model would then sort the items based on user preference resulting in a more personal experience.

## 3.7. Dynamic Graph Neural Networks for Sequential Recommendation[10]

Modeling user preference from historical sequences is one of the core problems of sequential recommendation. Existing methods in this field are widely distributed from conventional methods to deep learning methods. However, most of them only model users’ interests within their own sequences and ignore the dynamic collaborative signals among different user sequences, making it insufficient to explore users’ preferences.

The paper proposes a new method named DSGR which connects different user sequences through a dynamic graph structure exploring the interactive behavior of users and items with time and order information.

DSGR leverages the collaborative information among different user sequences not only each user individually.

For example, in the figure below at time user1 interacts with , and while also having high order connection with user2 and user3 we can use this connection in predicting user1’s sequence.

DSGR also considers the dynamic influence of the high-order collaboration information at different times, user1’s graph changes overtime.

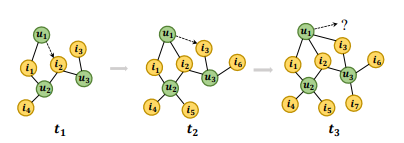


Figure ‑ DSGR graph[10]

The DSGR model’s architecture has 4 components:

1. Dynamic graph construction
2. Sub graph sampling
3. Dynamic graph recommendation networks
4. Predication layer

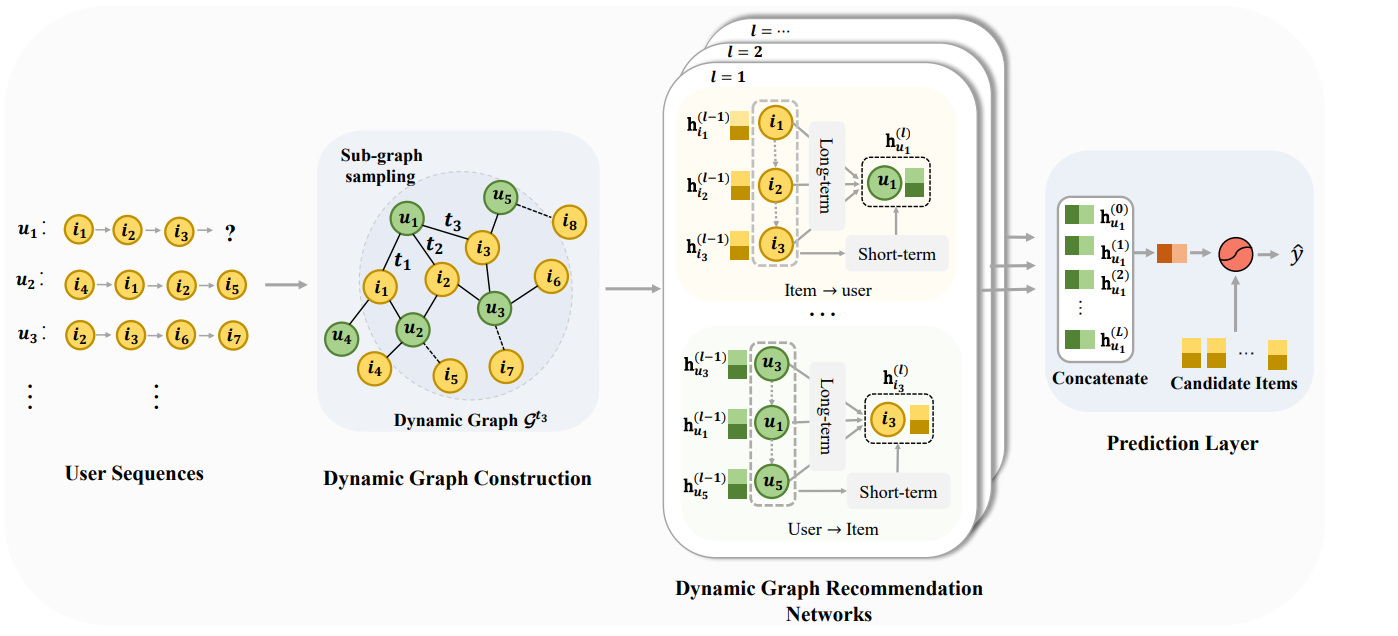
The graph is constructed in the following manner, when a user u acts on an item i at time an edge is established between u and i, in addition to the interaction time between users and items the graph constructed also records the order information between them making the graph more suitable for the sequential recommendation task.  
Predicting the next item in the predicted sequence is equivalent to predicting which item is linked on the user’s node in the graph.

Figure ‑ DSGR[10]

The next module in the architecture is the sub-graph sampling, as the number of the user sequence extends, the number of neighboring sequences increases and the scale of the dynamic graph composed of all users is also gradually expanding, all this increases the computational cost and introduces more noise to the graph to handle this issue a sampling algorithm is used.

A user node is chosen as the anchor node and select its most recent n first order neighbors from the graph where n is the maximum length of user sequence, for each item we use them as anchor node to sample the set of users who have interacted with them.

After sampling each sub-graph contains the nodes of the sequence and its associated sequences.

In the DGRN component consists of message propagation and node updating components, the message propagation mechanism aims to learn the information to propagate from user to item and from item to user.

Predicting the next item for a user is equivalent to predicting the link of user node u of a subgraph.

After acting the L layers on DGRN on graph, we obtain L embeddings of the user’s node, the user’s embedding in each layer emphasized various user preferences, these embeddings are concatenated to get final embedding for node.

To get score of each item a link function is defined as

Where is the user’s concatenated embedding and trainable transformation matrix.

**3.1. Comparative Study of Previous Work**

In this section give a comparative, classified short literature review of the latest publications the latest publications related to your project within past three years if applicable

**3.1. Implemented Approach**

Conclude this chapter by this section stating the approach chosen from those reviewed, **but more important your justification why you chose this approach** along with any modifications added to the approach.

Notice, you may be implementing several techniques however you must illustrate the general framework for your approach.

**Chapter 4: System Design and Architecture**

This chapter represents the main body of your project. It should describe the project in full details. This chapter should answer the questions: “what has been done?” and “how it has been done?”. As such, the steps you went through to realize the project should be highlighted and properly discussed. Your scientific approaches and methodologies should be clarified. The discussion should adopt a logical flow starting from the whole block diagram, to coarse modules, and finally to fine modules. While writing this chapter, try to give as much details as possible, such that an interested reader could easily replicate your work and improve it.

In this space, before the first section, write an introductory paragraph on how you design and build your project

**4.1. Overview and Assumptions**

In this section, introduce how you design you system and develop its underlying architecture. Any employed assumptions should be clearly enumerated and justified.

**4.2. System Architecture**

The architecture of your system should be given in this section. This architecture should be first represented as a block diagram (subsection 5.2.1), which clarifies different project modules and the connections between them. You may add more subsections to properly explain your design. If possible, flowcharts are better included to ensure that the big picture and the interaction between different modules are very clear to the reader. Thereafter, each module should have a separate subsequent section to clearly describe and discuss it.

Asd3 [11]

**4.2.1. Block Diagram**

Draw the block diagram of your architecture and generally discuss its modules. After reading this subsection, interested audience should have understood the big picture of your system design and architecture. The interaction between modules should also be conveyed in this subsection

Ad3[12]

**4.3. Module 1**

Each module within your architecture should have a distinct section to explain the design of the module itself. Again, give as much details as possible, so that the reader could easily understand how the module is designed and what are the constraints that affect its design?

**4.3.1. Functional Description**

Explain the functional description of the module

**4.3.2. Modular Decomposition**

Explain the modular decomposition of the coarse module into smaller fine ones

**4.3.3. Design Constraints**

Explain the constraints that affect the design of the module

**4.3.4. Other Description of Module 1**

Give any other necessary discussion of the module to ensure that it is clearly described.

**4.4. Module 2**

Each module within your architecture should have a distinct section to explain the design of the module itself. Again, give as much details as possible, so that the reader could easily understand how the module is designed and what are the constraints that affect its design?

**4.4.1. Functional Description**

Explain the functional description of the module

**4.4.2. Modular Decomposition**

Explain the modular decomposition of the coarse module into smaller fine ones

**4.4.3. Design Constraints**

Explain the constraints that affect the design of the module

**4.4.4. Other Description of Module 2**

Give any other necessary discussion of the module to ensure that it is clearly described.

**Chapter 5: System Testing and Verification**

In this chapter, you have to explain all the steps you carried out to ensure that project outcomes are realized correctly. Your testing setup, strategy and environment should therefore be described. Your efforts for unit testing as well as integrated system testing should be given. Finally, the results from different testing scenarios should be highlighted and discussed.

In this space, before the first section, write an introductory paragraph on how you test and verify the correct operation of your system

**5.1. Testing Setup**

Explain the setup you are using in testing your project

**5.2. Testing Plan and Strategy**

Explain the methodology you follow while testing your project in details

**5.2.1. Module Testing**

Explain the steps you carried out to test different modules within the project. Give and discuss the results obtained from the testing of these modules

**5.2.2. Integration Testing**

Explain the steps you carried out to test the integrated system of your project. Give and discuss the results obtained from this whole project testing

**5.3. Testing Schedule**

Mention your testing schedule

**5.4. Comparative Results to Previous Work**

Give a summary of comparative results to previous work in Tabulated and or Graphical form along with a short commentary.

**Chapter 6: Conclusions and Future Work**

This chapter should summarize the whole project, it features and limitation. Moreover, you should give directions for future work

In this space, before the first section, write an introductory paragraph for the chapter

**6.1. Faced Challenges**

Mention all the problems/challenges that you faced while working with the project and how you overcome them

**6.2. Gained Experience**

Mentioned the experience/skills that you gained from working with the project

**6.3. Conclusions**

Write your conclusions regarding the project. Mention its features and limitations

**6.4. Future Work**

Give possible extensions, enhancements and future work of you project, such that subsequent students could build on your work and develop larger systems/platforms.

**References**

The references should be ordered according to their appearance in the text. Ensure that all references are cited throughout your report text. The following are examples of how to write different types of references “[1] Book, [2] Journal/magazine articles, [3] conference paper, [4] website, [5] thesis”. Replace the fields with those of your used references. Question marks “??” should be replaced by the corresponding number

1. Author1, Author 2,…, “Book title,” name of publishing firm, edition, year
2. Author1, Author2,…., “Title of journal article,” name of the journal, vol. ??, no. ??, pp. ??, year of publication
3. Author1, Author2,…, “Title of conference paper,” in proceedings of conference name, city, country, date, year, pp. ??
4. Author or Corporation name, “Title,” year, link for the website, last accessed: date of last access
5. Author, “Thesis title,” M.Sc./Ph.D. thesis, Department, University, year

[1] “Global food trends 2021: How our habits have changed, as told by social images | YouScan.” https://youscan.io/blog/food-trends/ (accessed May 23, 2023).

[2] J. Cooper, “Cooking trends: The Digital Kitchen - Think with Google,” 2015.

[3] “basyl - your AI cooking assistant.” https://www.basyl.co/create/basic (accessed May 27, 2023).

[4] “khadija267/Recipes-Recommendation-Chatbot: A project about recommeing recipes and ingredients based on the needs of the user.” https://github.com/khadija267/Recipes-Recommendation-Chatbot (accessed May 27, 2023).

[5] “What is a Recommendation System? | Data Science | NVIDIA Glossary.” https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/ (accessed Jun. 01, 2023).

[6] “What is a Recommendation System? | Data Science | NVIDIA Glossary.” https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/ (accessed May 27, 2023).

[7] A. Vaswani *et al.*, “Attention Is All You Need,” Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.03762

[8] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” [Online]. Available: https://github.com/tensorflow/tensor2tensor

[9] T. Wang, Y. M. Brovman, and S. Madhvanath, “Personalized Embedding-based Recommendations at eBay,” in *IEEE International Conference on Program Comprehension*, IEEE Computer Society, 2022, pp. 36–47. doi: 10.1145/nnnnnnn.nnnnnnn.

[10] M. Zhang, S. Wu, X. Yu, Q. Liu, and L. Wang, “Dynamic Graph Neural Networks for Sequential Recommendation,” Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.07368

[11] E. Guo *et al.*, “Learning to Measure Change: Fully Convolutional Siamese Metric Networks for Scene Change Detection,” Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.09111

[12] L. Zhou, C. Xu, and J. J. Corso, “Towards automatic learning of procedures from web instructional videos,” in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, AAAI press, 2018, pp. 7590–7598. doi: 10.1609/aaai.v32i1.12342.

**Appendix A: Development Platforms**

**and Tools**

This appendix explains used tools, platforms, and hardware kits. Any ready-made module should be mentioned and discussed in this appendix. The appendix is divided into two main sections; one for the hardware and the other is for software. Within each section, you could add as much subsections as needed, according to the number of tools and platforms that you use in your project.

In this space, before the first section, write an introductory paragraph to the appendix

**A.1. Hardware Platforms**

A description of any used hardware platforms/kit should be written in this section. Each platform/kit is better described in a separate subsection. (A1.1..)

**A.2. Software Tools**

A description of any used software tool/package should be written in this section. Each tool/package is better described in a separate subsection (A2.1,..)

**Appendix B: Use Cases**

Include all your use cases

**Appendix C: User Guide**

Prepare a user guide for your project. Ensure that the guide is clear, detailed and easy for an ordinary customer to use your project. Employ figures and charts as needed to facilitate the use of your guide

**Appendix D: Code Documentation**

Your code or parts of the code you feel necessary could be included here (optional) however for one copy of this report an attached CD with all of the code is a must.

Remember you will deliver three copies of this report.

**Appendix D: Feasibility Study**

Give a detailed feasibility study of your project