Knowledge Graph Attention Network Review

Babiński Michał
Dymanowski Krzysztof
Wesołowski Jędrzej
Gdańsk University of Technology
TODO

[1]Gdansk University of Technology
[2]Electronics and Information Science Department

Abstract

Knowledge Graph Attention Network (KGAT) to system rekomendacyjny który łączy interakcje user-item (użytkownik-przedmiot) z bogatymi informacjami kontekstowymi z grafów wiedzy (KG). W przeciwieństwie do tradycyjnych metod, KGAT nie traktuje interakcji jako odizolowanych zdarzeń. Zamiast tego wychwytuje zależności wyższego rzędu między przedmiotami a ich atrybutami w połączonej strukturze grafu wiedzy i grafu użytkownik-przedmiot. Dzięki zastosowaniu propagacji osadzeń i mechanizmów uwagi, KGAT identyfikuje istotne powiązania, oferując bardziej precyzyjne, zróżnicowane i wytłumaczalne rekomendacje.

W tym raporcie omawiamy, jak zmodernizować framework KGAT, wykorzystując najnowsze technologie w dziedzinie uczenia maszynowego opierającego się na grafach i umożliwiając uczenie modelu na maszynach GPU, co powinno wielokrotnie przyspieszyć uczenie. Dodatkowo testujemy jego działanie na nowym zestawie danych, aby sprawdzić, jak dobrze dostosowuje się i poprawia w różnych scenariuszach. Te eksperymenty mają za zadanie dać głębszy wgląd w to, jak te ulepszenia i różnice w danych wpływają na dokładność rekomendacji oraz zachowanie ich interpretowalności.

1 Wyniki eksperymentów

W ramach projektu przeprocesowano zbiór danych amazon-kindle (https://www.kaggle.com/datasets/asaniczka/amazon-kindle-books-dataset-2023-130k-books) do formatu przyjmowanego przez algorytm, oraz powtórzono eksperymenty zmieniając nastawienia hiperparametrów. Ze względu na olbrzymie wymagania obliczeniowe, trening modeli przeprowadzany był przez 50 epok, zamiast domyślnego tysiąca epok które zaproponowali autorzy. Ponadto dokonano ewaluacji na przygotowanym zbiorze.

Proces tworzenia zbioru danych:

1. Zbiór danych nie zawierał interakcji użytkownik-przedmiot, zatem trzeba było przygotować skrypt który tworzył plik .txt w formacie przyjmowanym przez implementację
2. Najpierw oczyszczono zbiór tak, aby zawierał użytkowników którzy wystawili co najmniej 5 ocen.
3. Na podstawie tego utworzono zbiór interakcji użytkownik - przedmiot

```
# user_id item_id1 item_id2 item_id3 ...
0 10 20 30
1 15 25
2 10 25 35
```

4. Ostatnim krokiem było przygotowanie grafu wiedzy, który zawierał trójki relacji w następujący sposób
5. W tym momencie posiadamy główne pliki train.txt, test.txt oraz kg_final.txt które są akceptowane przez algorytm.

```
# (Książka 0, Napisana przez, Autor 100)
0 0 100
# (Książka 0, Sprzedana przez, Amazon.com Servic
0 1 200
# (Książka 0, Należy do kategorii, Rodzicielstwo
0 2 300
# (Książka 0, Opublikowana w, 2015)
0 3 2015
```

Tabele wynikowe:

Rozmiar embeddingów 32, learning rate 1e-4, bez pre-treningu: Rozmiar embeddingów 32, learning rate 1e-5, bez pre-treningu:

Rozmiar embeddingów 32, learning rate 1e-5, bez pre-treningu:

Dla zbioru amazon-kindle wyniki prezentują się nastepująco: Rozmiar embeddingów 32, learning rate 1e-4, bez pre-treningu:

Największy problem sprawiało utworzenie samego zbioru danych, gdyż wymagało ono od nas opracowania grafu wiedzy. Ponadto, ontologia tego grafu nie opierała się na Freebase, tak jak w przypadku pozostałych zbiorów danych, więc nie jesteśmy do końca pewni jej poprawności. Największym

Table 1: *Amazon-book*

Metric	10 Epochs	20 Epochs	30 Epochs	40 Epochs	50 Epochs
Precision@20	0.0043	0.0047	0.0053	0.0061	0.0065
Recall@20	0.0386	0.0414	0.0466	0.0549	0.0586
NDCG@20	0.0184	0.0190	0.0222	0.0261	0.0280
Precision@40	0.0036	0.0041	0.0045	0.0051	0.0054
Recall@40	0.0629	0.0711	0.0792	0.0906	0.0954
NDCG@40	0.0245	0.0265	0.0304	0.0351	0.0373
Precision@60	0.0032	0.0037	0.0040	0.0046	0.0048
Recall@60	0.0820	0.0937	0.1034	0.1190	0.1257
NDCG@60	0.0287	0.0316	0.0358	0.0414	0.0440
Precision@80	0.0029	0.0034	0.0037	0.0042	0.0044
Recall@80	0.1003	0.1144	0.1267	0.1439	0.1512
NDCG@80	0.0325	0.0358	0.0406	0.0465	0.0492
Precision@100	0.0027	0.0032	0.0035	0.0039	0.0041
Recall@100	0.1170	0.1342	0.1477	0.1652	0.1741
NDCG@100	0.0357	0.0396	0.0447	0.0507	0.0537

Table 2: *Yelp2018*

Metric	10 Epochs	20 Epochs	30 Epochs	40 Epochs	50 Epochs
Precision@20	0.0062	0.0087	0.0102	0.0115	0.0125
Recall@20	0.0122	0.0183	0.0217	0.0251	0.0271
NDCG@20	0.0105	0.0155	0.0185	0.0212	0.0231
Precision@40	0.0055	0.0077	0.0089	0.0099	0.0107
Recall@40	0.0200	0.0298	0.0353	0.0398	0.0429
NDCG@40	0.0131	0.0193	0.0229	0.0260	0.0282
Precision@60	0.0052	0.0070	0.0081	0.0090	0.0096
Recall@60	0.0279	0.0391	0.0457	0.0513	0.0549
NDCG@60	0.0156	0.0224	0.0263	0.0299	0.0322
Precision@80	0.0049	0.0066	0.0076	0.0083	0.0089
Recall@80	0.0346	0.0474	0.0553	0.0614	0.0655
NDCG@80	0.0177	0.0251	0.0294	0.0331	0.0356
Precision@100	0.0047	0.0062	0.0071	0.0078	0.0085
Recall@100	0.0403	0.0550	0.0638	0.0701	0.0747
NDCG@100	0.0195	0.0275	0.0321	0.0359	0.0385

Table 3: *Last-FM*

Metric	10 Epochs	20 Epochs	30 Epochs	40 Epochs	50 Epochs
Precision@20	0.0051	0.0067	0.0076	0.0081	0.0088
Recall@20	0.0183	0.0256	0.0296	0.0323	0.0348
NDCG@20	0.0114	0.0157	0.0182	0.0202	0.0217
Precision@40	0.0047	0.0061	0.0068	0.0073	0.0078
Recall@40	0.0346	0.0462	0.0528	0.0564	0.0609
NDCG@40	0.0166	0.0224	0.0256	0.0280	0.0301
Precision@60	0.0045	0.0057	0.0063	0.0067	0.0072
Recall@60	0.0499	0.0647	0.0730	0.0777	0.0834
NDCG@60	0.0210	0.0277	0.0314	0.0341	0.0365
Precision@80	0.0043	0.0054	0.0059	0.0063	0.0067
Recall@80	0.0643	0.0815	0.0909	0.0966	0.1038
NDCG@80	0.0248	0.0321	0.0361	0.0390	0.0419
Precision@100	0.0042	0.0052	0.0057	0.0060	0.0063
Recall@100	0.0778	0.0975	0.1078	0.1147	0.1219
NDCG@100	0.0282	0.0361	0.0404	0.0436	0.0464

Table 4: *Amazon-kindle*

Metric	10 Epochs	20 Epochs	30 Epochs	40 Epochs	50 Epochs
Precision@20	0.0059	0.0049	0.0070	0.0073	0.0059
Recall@20	0.0340	0.0211	0.0184	0.0212	0.0204
NDCG@20	0.0210	0.0179	0.0110	0.0178	0.0224
Precision@40	0.0061	0.0048	0.0083	0.0074	0.0056
Recall@40	0.0424	0.0271	0.0274	0.0267	0.0262
NDCG@40	0.0237	0.0218	0.0138	0.0230	0.0271
Precision@60	0.0050	0.0041	0.0077	0.0083	0.0048
Recall@60	0.0540	0.0314	0.0307	0.0343	0.0331
NDCG@60	0.0290	0.0256	0.0171	0.0295	0.0302
Precision@80	0.0054	0.0040	0.0075	0.0086	0.0057
Recall@80	0.0596	0.0395	0.0345	0.0441	0.0415
NDCG@80	0.0375	0.0292	0.0205	0.0329	0.0344
Precision@100	0.0062	0.0040	0.0061	0.0075	0.0046
Recall@100	0.0681	0.0482	0.0443	0.0489	0.0524
NDCG@100	0.0426	0.0324	0.0237	0.0384	0.0428

ograniczeniem zaś był czas treningu modelu, dla 50 epok trening jednego modelu na cPU trwał około 20 godzin na stacji roboczej (z procesorem i7 14700). Po modernizacji kodu i środowiska tak, aby korzystać z GPU, czas treningu dla 50 epok skrócił się do około 2,5 godziny.

2 Introduction

3 Opis metody

Sieci grafowe, takie jak Knowledge Graph Attention Network, wykorzystują graf wiedzy do wzbogacenia systemów rekomendacyjnych. W tradycyjnych metodach, takich jak filtrowanie kolaboracyjne czy modele nadzorowanego uczenia, interakcje użytkownika z przedmiotami traktowane są jako niezależne zdarzenia, co ogranicza zdolność do wykorzystania informacji kontekstowych takich jak atrybuty przedmiotów, profile użytkowników itp.

KGAT wprowadza innowacyjne podejście, łącząc graf wiedzy z grafem użytkownik-przedmiot w jedną hybrydową strukturę - graf wiedzy kolaboracyjnej (CKG). Istotnym jego elementem jest modelowanie relacji wyższego rzędu, które uwzględniają pośrednie powiązania między użytkownikami a przedmiotami poprzez wspólne atrybuty (na przykład wspólny reżyser filmów któe można zarekomendować innemu użytkownikowi na bazie historii oglądania innego użytkownika). Aby to osiągnąć, KGAT stosuje: Rekurencyjną propagację osadzeń(embeddings), w ramach któej osadzenia każdego węzła są aktualizowane na podstawie informacji od sąsiadów, co pozwala wydajnie uchwycić relacje wyższego rzędu, oraz agregację opartą na uwadze (attention), czyli attention mechanism który pozwala modelowi przypisywać różne wagi sąsiadom, uwzględniając wagę ich istotności w kontekście rekomendacji. Zalety sieci grafowych:

- Efektywne wykorzystanie informacji dodatkowych: Sieci grafowe skutecznie integrują cechy przedmiotów, profile użytkowników i inne dane kontekstowe, co poprawia wyniki rekomendacji.
- Modelowanie relacji wyższego rzędu: KGAT potrafi uchwycić dalekie powiązania między węzłami, co jest problematyczne dla tradycyjnych metod.
- Interpretowalność: Mechanizm uwagi pozwala interpretować, które relacje miały największy wpływ na wyniki modelu.
- Brak potrzeby ręcznego definiowania ścieżek: W przeciwieństwie do metod opartych na

ścieżkach, KGAT nie wymaga ręcznego definiowania ścieżek, co oszczędza czas i zasoby.

Wady sieci grafowych:

- Wysokie wymagania obliczeniowe: Liczba węzłów i relacji rośnie eksponencjalnie wraz ze wzrostem złożoności grafu, co może być problematyczne dla większych problemów.
- Złożoność modelowania relacji: Relacje wyższego rzędu wnoszą różny wkład do rekomendacji, co wymaga precyzyjnego ważenia i selekcji.
- Brak interpretowalności w niektórych modelach: Chociaż KGAT stawia na interpretowalność, inne podejścia (np. regularizacyjne) mogą być trudniejsze do zrozumienia.

4 Opis architektury

4.1 Szczegółowy opis algorytmu

Algorytm KGAT składa się z trzech głównych komponentów:

- Warstwa osadzania: Parametryzuje każdy węzeł jako wektor, zachowując strukturę grafu wiedzy.
- Warstwy propagacji osadzania z uwzględnieniem uwagi: Rekurencyjnie propagują osadzenia od sąsiadów węzła, aktualizując jego reprezentację i ucząc się wagi każdego sąsiada podczas propagacji.
- Warstwa predykcji: Agreguje reprezentacje użytkownika i przedmiotu ze wszystkich warstw propagacji i zwraca przewidywany wynik dopasowania.

4.2 Parametry Algorytmu

Wśród parametrów modelu możemy wyróżnić hiperparametry, w tym przypadku wymiar warstwy osadzeń i liczba wartsw propagacji. Dodatkowo algorytm dynamicznie wybiera rozmiary osadzeń w zależności od węzłów, co pozwala na bardziej elastyczne modelowanie relacji oraz wagi warstwy attention (warstwy uwagi. wagi uwagi brzmiały na tyle głupio że wolę używeać angielskiego).

4.3 Format danych wejściowych

Danymi wejściowymi grafu są trójki relacji między węzłami, w formacie (h, r, t), gdzie h to węzeł początkowy, r to relacja między węzłami, a t to węzeł docelowy.
Danymi wyjściowymi są przewidywane wyniki dopasowania użytkownika do przedmiotu.

4.4 Warstwy osadzania i atencji

Warstwa osadzania Knowledge Graph Attention Network (KGAT) wykorzystuje warstwę embeddingu do reprezentowania encji i relacji w grafie wiedzy jako wektorów liczbowych. Model ten bazuje na metodzie osadzania grafów wiedzy TransR, która umożliwia modelowanie semantyki relacji między encjami.

Osadzanie encji i relacji Każda encja h, t i relacja r w grafie wiedzy jest reprezentowana jako wektor w przestrzeni wektorowej. Przekształcanie osadzeń odbywa się poprzez macierz transformacji W_r, która rzutuje encje do przestrzeni specyficznej dla danej relacji:

$$e_r(h) + e_r \approx e_r(t), \tag{1}$$

co oznacza, że wektor encji początkowej powinien być bliski wektorowi encji docelowej po zastosowaniu transformacji relacyjnej.

Funkcja kosztu W celu optymalizacji osadzeń KGAT minimalizuje funkcję kosztu, opartą na rangowej stracie logistycznej:

$$L_{KG} = \sum_{(h,r,t,t')} -\ln \sigma\left(g(h,r,t') - g(h,r,t)\right), \tag{2}$$

co pozwala na skuteczne rozróżnianie prawdziwych i fałszywych relacji w grafie.

Warstwy atencji i propagacji embeddingu Aby efektywnie modelować zależności wysokiego rzędu w grafie wiedzy, KGAT stosuje propagację embeddingów inspirowaną sieciami grafowymi (Graph Neural Networks, GNN) oraz mechanizmem atencji (attention mechanism).

Propagacja informacji Dla każdej encji h, nowa reprezentacja jest wyliczana na podstawie jej sąsiednich węzłów t oraz relacji r. Wartość propagacji kontroluje współczynnik tłumienia $\pi(h,r,t)$, który określa ilość informacji przekazywanej z węzła t do h:

$$e_{N_h} = \sum_{(h,r,t)\in N_h} \pi(h,r,t) e_t. \tag{3}$$

Mechanizm atencji Aby poprawić jakość rekomendacji, KGAT wykorzystuje mechanizm atencji do dynamicznego przypisywania różnych wag do sąsiednich węzłów w trakcie propagacji embeddingów. Waga relacji $\pi(h,r,t)$ jest obliczana jako:

$$\pi(h,r,t) = (W_r e_t)^\top \tanh(W_r e_h + e_r), \tag{4}$$

po czym wartości te są normalizowane za pomocą funkcji softmax:

$$\pi(h,r,t) = \frac{\exp(\pi(h,r,t))}{\sum_{(h,r',t')\in N_h} \exp(\pi(h,r',t'))}. \quad (5)$$

Dzięki temu model może selektywnie wzmacniać istotne zależności między encjami.

Agregacja informacji Ostateczna reprezentacja węzła h jest sumą jego oryginalnej reprezentacji oraz propagowanych embeddingów, co można zapisać jako:

$$e_h^{(1)} = f(e_h, e_{N_h}). \quad (6)$$

KGAT testuje różne strategie agregacji, w tym:

Wielowarstwowa propagacja Model KGAT może iteracyjnie propagować embeddingi przez wiele warstw, co pozwala uchwycić zależności wysokiego rzędu:

$$e_h^{(l)} = f(e_h^{(l-1)}, e_{N_h}^{(l-1)}). \quad (7)$$

Zastosowanie kilku warstw umożliwia modelowi uwzględnienie relacji o wyższych rzędach, co poprawia jakość rekomendacji.

Podsumowanie Warstwy embeddingu i atencji w KGAT umożliwiają efektywne modelowanie wysokorzędowych zależności w grafie wiedzy. Mechanizm atencji pozwala na selektywne ważenie informacji od sąsiadujących węzłów, a propagacja embeddingów umożliwia hierarchiczne przekazywanie informacji. Dzięki temu KGAT osiąga lepsze wyniki w rekomendacjach niż wcześniejsze metody, jednocześnie oferując lepszą interpretowalność wyników.

4.5 Uniwersalność

Algorytm może być stosowany w różnych domenach, ale wymaga odpowiedniego grafu wiedzy który trzeba przygotować przed procesem uczenia. Dodatkowo jego złożoność obliczeniowa sprawia, że może być problematyczny bądź nawet niemożliwy dla większych zbiorów danych.

5 Przegląd metod rekomendacyjnych

Systemy rekomendacyjne stanowią kluczowy element nowoczesnych aplikacji internetowych, umożliwiając personalizację treści na podstawie analizy zachowań użytkowników. W niniejszej sekcji przedstawiono główne podejścia stosowane w systemach rekomendacyjnych wraz z ich zaletami i ograniczeniami.

5.1 Filtracja kolaboratywna

Filtracja kolaboratywna (ang. *Collaborative Filtering, CF*) opiera się na analizie wzorców zachowań użytkowników w celu przewidywania ich preferencji. Wyróżnia się dwa główne podejścia:

- *User-based CF* – rekomendacje są generowane na podstawie podobieństwa między użytkownikami,
- *Item-based CF* – rekomendacje bazują na podobieństwie między przedmiotami.

Zaletą metody jest jej niezależność od treści przedmiotów, jednak problemem pozostaje efekt zimnego startu oraz skomplikowana skalowalność w przypadku dużych zbiorów danych.

5.2 Dekompozycja macierzy (Matrix Factorization)

Dekompozycja macierzy (ang. *Matrix Factorization, MF*), np. metoda SVD (ang. *Singular Value Decomposition*) lub ALS (ang. *Alternating Least Squares*), redukuje wymiarowość danych poprzez reprezentację interakcji użytkownik-przedmiot w przestrzeni ukrytych cech. Pozwala to na identyfikację wzorców preferencji, lecz metoda ta nadal cierpi na problem zimnego startu i słabo modeluje skomplikowane zależności.

5.3 Filtracja oparta na treści

Podejście oparte na treści (ang. *Content-Based Filtering*) analizuje cechy przedmiotów (np. gatunek filmu, słowa kluczowe w artykule) w celu przewidywania preferencji użytkownika. Metoda ta dobrze sprawdza się w przypadku nowych użytkowników, jednak może prowadzić do efektu *filter bubble*, czyli zbyt homogenicznych rekomendacji.

5.4 Metody wykorzystujące głębokie sieci neuronowe

Głębokie sieci neuronowe (ang. *Deep Learning-based Recommendations*), np. *Neural Collaborative Filtering (NCF)* lub autoenkodery, pozwalają na modelowanie nieliniowych zależności w danych. Chociaż zapewniają wysoką jakość rekomendacji, wymagają znacznych zasobów obliczeniowych oraz dużych zbiorów danych do skutecznego działania.

5.5 Rekomendacje oparte na grafach

Metody oparte na grafach (ang. *Graph-Based Recommendations*), takie jak *Graph Neural Networks (GNN)*

czy *Knowledge Graph Attention Network (KGAT)*, modelują relacje między użytkownikami i przedmiotami w postaci struktur grafowych. Takie podejście umożliwią lepsze uchwycenie kontekstowych zależności, lecz ich implementacja jest kosztowna obliczeniowo.

5.6 Podejścia hybrydowe

Hybrydowe systemy rekomendacyjne łączą powyższe podejścia w celu uzyskania bardziej precyzyjnych wyników. Przykładem jest system rekomendacyjny Netflixa, który integruje filtrację kolaboratywną, analizę treści oraz uczenie głębokie. Choć hybrydowe podejścia są bardzo skuteczne, ich implementacja jest złożona i wymaga znacznych zasobów obliczeniowych.

5.7 Podsumowanie

Tabela 5 przedstawia porównanie kluczowych metod rekomendacyjnych pod kątem zalet i ograniczeń.

6 Porównanie istniejących technologii

7 Task 3

W trzecim etapie projektu studenci, korzystając z wiedzy zdobytej w poprzednich dwóch etapach projektu, proponują własną modyfikację algorytmu i/lub eksperymentu przeprowadzonego przez autorów oryginalnego artykułu i planują eksperyment mający na celu weryfikację zaproponowanej modyfikacji. Na początku studenci powinni przemyśleć możliwe modyfikacje, które zidentyfikowali w poprzednich sprawozdaniach. Można też rozszerzyć pulę o nowe propozycje. Studenci powinni opisać wszystkie zidentyfikowane przez siebie możliwe modyfikacje, zarówno samego algorytmu, jak i eksperymentu (nie obliguje to do implementacji wszystkich rozszerzeń w kolejnym etapie). Podczas opisu każdej z możliwych modyfikacji algorytmu i/lub eksperymentu, studenci powinni rozważyć następujące kwestie:

- Na czym polega dana modyfikacja?
- W jaki sposób zmieni ona sposób wykonania algorytmu/eksperymentu?
- W którym miejscu w kodzie będą wprowadzone zmiany?
- Jakie są motywacje wprowadzenia danej modyfikacji?
- Czy pozwoli ona ulepszyć algorytm/eksperyment? W jaki sposób?
- W przypadku modyfikacji eksperymentu powinno się rozważyć również pytania:
  - Jaką nową wiedzę o metodzie możemy zyskać dzięki danej modyfikacji eksperymentu?
  - Do czego w praktyce może się ta wiedza przydać?

W ostatniej części etapu trzeciego studenci wybierają dwie modyfikacje i planują dla nich eksperymenty. Należy również uzasadnić wybór modyfikacji do implementacji. Plan eksperymentu powinien zawierać informacje o tym, co będzie badane i w jaki sposób. Studenci powinni zidentyfikować wszystkie zmienne zależne i niezależne, w tym również hiperparametry wymagające wcześniejszego strojenia. Każdy krok eksperymentu powinien być dokładnie zaplanowany i opisany. Ważny jest również wybór i opis zbiorów danych do eksperymentu.

7.1 Prezentacja do Etapu III

Prezentacja z Etapu III powinna zawierać podstawowe informacje o autorach, tj. imiona, nazwiska i numery indeksów wszystkich członków zespołu projektowego oraz informacje o analizowanym artykule, tj. nazwiska autorów i tytuł. Poniżej podano wymagane elementy merytoryczne sprawozdania. Prezentacja powinna trwać 10-12 minut.

1. Możliwe modyfikacje
   - Dla każdej zidentyfikowanej możliwej modyfikacji, studenci powinni rozważyć następujące elementy:
     (a) Istota modyfikacji
       - Typ modyfikacji (algorytmu czy eksperymentu).
       - Na czym polega dana modyfikacja?
       - W jaki sposób zmieni ona sposób wykonania algorytmu/eksperymentu?
       - W którym miejscu w kodzie będą wprowadzone zmiany?

     (b) Motywacje do wprowadzenia danej modyfikacji
       - Jakie są motywacje wprowadzenia danej modyfikacji?
       - Czy pozwoli ona ulepszyć algorytm/eksperyment? W jaki sposób?
       - Opcjonalne (tylko w przypadku modyfikacji eksperymentu): Jaką nową wiedzę o metodzie możemy zyskać dzięki danej modyfikacji eksperymentu? Do czego w praktyce może się ta wiedza przydać?

Metoda	Zalety	Ograniczenia
Filtracja kolaboratywna	Wysoka skuteczność przy dużych danych	Problem zimnego startu, skalowalnoś
Matrix Factorization	Kompaktowa reprezentacja danych	Nie modeluje skomplikowanych zależn
Filtracja oparta na treści	Nie wymaga dużych zbiorów użytkowników	Filter bubble, wymaga cech przedmiot
Deep Learning	Wysoka jakość rekomendacji	Wymaga dużych danych, koszt obliczeni
Graph-Based	Modeluje złożone relacje	Kosztowna implementacja
Podejścia hybrydowe	Najlepsza jakość rekomendacji	Złożona integracja, wysokie wymagania obli

Table 5: *Porównanie metod rekomendacyjnych*

2. Plan eksperymentu
   (a) Wybór modyfikacji
      - Która z powyższych modyfikacji została wybrana do implementacji? Dlaczego akurat ta?

   (b) Opis eksperymentu
      - Dokładny opis planowanego eksperymentu.
      - Jakie zbiory danych zostaną wykorzystane?
      - Jakie kroki zostaną przeprowadzone?
      - Co będziemy badać?
      - Jakie będą zmienne zależne a jakie niezależne w projektowanym eksperymencie?
      - Czy będą wykorzystywane miary jakości systemów rekomendacyjnych? Jeśli tak, to jakie?
      - Czy wymagane jest wcześniejsze strojenie hiperparametrów modelu? Jeżeli tak, to których i w jaki sposób będzie wykonane?

8 Task 4

W czwartym etapie projektu studenci, korzystając z planu eksperymentu zaprojektowanego w trzecim etapie, przeprowadzają eksperyment naukowy, analizują jego wyniki i formułują wnioski. W pierwszej części czwartego etapu studenci powinni wykonać dwa eksperymenty zgodnie z planem opisanym w sprawozdaniu do etapu trzeciego. Istotna jest tutaj zgodność z opisanym planem. Jeżeli wystąpią jakiekolwiek trudności z realizacją zaplanowanych w eksperymencie czynności, studenci powinni opisać te problemy wraz z rozwiązaniem, które zostało zastosowane. Wszelkie modyfikacje planu również powinny zostać skrupulatnie opisane. Po wykonaniu każdego z eksperymentów, studenci powinni zebrać otrzymane wyniki w wygodnej do analizy formie, np. wykres, tabela. Dobrze by było, gdyby zostały przedstawione w takiej formie, która umożliwia porównanie otrzymanych wyników z tymi z oryginalnego artykułu (w uzasadnionych przypadkach nie jest to konieczne). Otrzymane wyniki powinny zostać przeanalizowane samodzielnie, jak również porównane z tymi z oryginalnego artykułu (o ile to możliwe). Na podstawie planu modyfikacji i wyników oraz oryginalnego artykułu, studenci powinni wyciągnąć wnioski z przeprowadzonych badań (dla każdego eksperymentu osobno). W ostatniej części etapu czwartego studenci podsumowują pracę ze wszystkich dotychczasowych etapów i wyciągają wnioski końcowe. Kluczowe jest zidentyfikowanie, czy podczas przeprowadzonych eksperymentów powstała nowa wiedza o badanym algorytmie (i opisanie jej, jeśli tak). Możliwe jest tu uwzględnienie wszelkich uwag dotyczących oryginalnej metody, procedury ewaluacji, zaimplementowanych modyfikacjach i otrzymanych wyników.

8.1 Prezentacja do Etapu IV

Prezentacja z Etapu IV powinna zawierać podstawowe informacje o autorach, tj. imiona, nazwiska i numery indeksów wszystkich członków zespołu projektowego oraz informacje o analizowanym artykule, tj. nazwiska autorów i tytuł. Poniżej podano ramowe elementy merytoryczne sprawozdania.

1. Analizowany algorytm
   - Krótkie przypomnienie, co robi analizowany algorytm.

2. Eksperymenty
   (a) Przebieg eksperymentu
      - Dokładny opis przeprowadzonego eksperymentu.
      - Czy udało się przeprowadzić modyfikację zgodnie z założonym planem? Jeśli nie, to co się zmieniło?
      - Czy wystąpiły jakieś niespodziewane problemy? Jeśli tak, to jakie?

   (b) Wyniki
      - Dokładny opis otrzymanych wyników z eksperymentu wraz z tabelami i/lub wykresami.

   (c) Porównanie wyników i wnioski

- Czy wyniki różnią się od wyników otrzymanych przez autorów oryginalnego artykułu? Jeśli tak, to czy są lepsze czy gorsze? Jak myślisz, dlaczego tak jest?
- Jeśli modyfikacją było nowe zastosowanie metody, to czy po przeprowadzonym eksperymencie możemy uznać, że metoda sprawdzi się w tym zastosowaniu? Dlaczego? Itd.

3. Wnioski końcowe
    - Co możesz powiedzieć o oryginalnym algorytmie i własnych modyfikacjach po przeprowadzeniu eksperymentów?
    - Czy możesz wysnuć jakieś wnioski z wszystkich eksperymentów (z artykułu i własnych)?
    - Czy powstała nowa wiedza podczas realizacji projektu? Jeśli tak, to jaka? Itp.

9 Task 5

9.1 Przebieg Etapu V

W piątym etapie projektu studenci przygotowują artykuł naukowy zgodnie z załączonym szablonem. Artykuł powinien zawierać wszystkie kluczowe informacje dotyczące wszystkich wcześniejszych etapów projektu, ze szczególnym naciskiem na etap czwarty (własny eksperyment i wnioski).

9.2 Sprawozdanie do Etapu V

Sprawozdanie z Etapu V będzie stanowił artykuł naukowy, który powinien zawierać podstawowe informacje o autorach, tj. imiona i nazwiska wszystkich członków zespołu projektowego oraz informacje o analizowanym artykule, tj. nazwiska autorów i tytuł. Ponadto, artykuł powinien zawierać następujące elementy:

1. Wprowadzenie do problematyki.
2. Przegląd stanu wiedzy/istniejących modyfikacji algorytmu
3. Analizowany algorytm
4. Przeprowadzone eksperymenty wraz z wynikami
5. Wnioski końcowe

Dokładny dobór treści w ramach punktów jest pozostawiony studentom i podlega ocenie. Artykuł powinien być napisany w języku angielskim z wykorzystaniem załączonego szablonu LaTeX. Długość artykułu powinna wynosić 10-14 stron (łącznie z bibliografią).

9.3 Forma i sposób zaliczenia Etapu V

Za etap można otrzymać maksymalnie 8 punktów. Do zaliczenia zadania wymagane jest otrzymanie minimum 4 punkty. Oceniany będzie dobór i sposób zaprezentowania treści. Termin wgrania artykułów upływa 21.01.2024 o godz. 23:59.