Data Mining

Final Project: News Recommendation Prediction

TA: Hank Chen

h7a4n1k.cs12@nycu.edu.tw

Task Description

Topic: News Recommendation Prediction

You are given several entries of user interaction history, you need to use these information to predict whether a user will click on a news or not.

Goal: Use **any model or method** to predict whether a user will click on a news.

1. Submit a report and source code to E3
2. Upload your prediction to Kaggle competition

Dataset

You are given 2 datasets (download via Kaggle), **train** and **test**, each with 3 files:

- behaviors.tsv
- entity_embedding.vec
- news.tsv

Dataset (cont.)

- behaviors.tsv
  - **user_id**: user id.
  - **time**: the time the user visited the website homepage.
  - **clicked_news**: **space-separated list** of user's click history.
  - **impressions**: **space-separated list (15 items)**, whether user clicked on the recommended news or not. (Format: {news_id}-{clicked}, clicked is 0 or 1.)

Dataset (cont.)

1	2	3	4	5
id	user_id	time	clicked_news	impressions
0	U1349561	11/11/2019 9:31:08 AM	N410559 N109405 N79284 N812877 N311012 N362	N383574-0 N727666-0 N169045-0 N846428-0 N703824-0 N2970-0 N751592-0 N159567-0 N
1	U2788121	11/9/2019 9:13:19 AM	N642952 N253717 N857922 N684266 N291776 N12	N184823-0 N107900-0 N29500-0 N122187-0 N487132-0 N811701-0 N49569-0 N460476-0 N
2	U686145	11/12/2019 6:21:28 AM	N900496 N118253 N510477 N167498 N693772	N499466-0 N665940-0 N394508-1 N386423-0 N396755-0 N88097-0 N489400-0 N624080-0
3	U2794941	11/13/2019 9:30:05 AM	N417895	N96322-0 N909778-0 N656987-0 N594694-0 N306259-1 N820689-0 N602779-0 N374313-0
4	U1838845	11/10/2019 5:03:16 AM	N187833 N272183 N482344 N65242 N211696 N194	N27904-0 N731054-0 N281766-0 N177459-0 N768567-0 N530889-0 N837454-0 N647236-0
5	U137749	11/9/2019 4:06:02 PM	N522916	N537665-0 N367040-0 N219690-0 N407682-0 N503484-0 N627384-0 N103931-0 N225458-0
6	U1957641	11/14/2019 11:34:50 PM	N271189 N187833 N232675 N746132 N7058 N8525	N577516-0 N453490-1 N910527-0 N830356-0 N30641-0 N520802-0 N618298-0 N860603-0
7	U17537	11/13/2019 9:08:41 AM	N436802 N93956 N24040 N501650 N315863 N4419	N668670-0 N619299-0 N71990-0 N508258-0 N88762-0 N656987-0 N341483-0 N894847-0 N
8	U436289	11/9/2019 10:47:35 AM	N716914 N887847 N603640 N394312	N370435-0 N460476-1 N302297-0 N780845-0 N103931-0 N48050-0 N44046-0 N534683-0 N
9	U1166697	11/12/2019 10:23:09 AM	N188540 N520977 N712420 N536321 N719196 N40	N859728-0 N379668-0 N272309-0 N447659-0 N291965-0 N866063-0 N767489-1 N96322-0
10	U2617573	11/11/2019 11:37:41 AM	N117280 N324172 N794824 N284391 N279442 N22	N796182-1 N479929-0 N69540-0 N751592-0 N881610-0 N820815-0 N386423-0 N183402-0
11	U1714313	11/14/2019 7:50:07 PM	N266681 N446056 N871166 N610647 N272183 N41	N54091-0 N25356-0 N380571-1 N678701-0 N202022-0 N483394-0 N592013-0 N415011-0 N
12	U768033	11/13/2019 5:13:51 AM	N366900 N50633 N407535 N407535	N677098-0 N84275-0 N567870-0 N661999-0 N457186-0 N324900-0 N909778-1 N249559-0
13	U437037	11/11/2019 11:21:30 AM	N118253 N228510 N709634 N247291 N99675 N274	N772382-0 N894693-0 N614980-0 N735926-1 N563586-0 N374607-1 N258330-0 N589164-0
14	U2195893	11/14/2019 6:38:23 AM	N290383 N283117 N863760 N54189 N9767 N5010	N821627-0 N32531-1 N290656-0 N789280-0 N756975-0 N515321-0 N501377-0 N392324-0
15	U2384397	11/13/2019 4:32:24 AM	N247529 N76029 N37130 N7891 N5350 N614294 N	N71759-0 N324900-0 N567870-0 N194161-1 N909778-0 N96322-0 N36150-0 N791814-0 N8
16	U2650561	11/10/2019 4:15:22 PM	N136796	N97799-0 N839519-0 N315863-0 N750332-0 N184571-0 N270741-0 N114816-0 N545932-0
17	U760757	11/13/2019 2:42:03 PM	N342792 N900496 N77849 N298916 N391316 N38	N906565-0 N161534-0 N13456-0 N536734-0 N872734-1 N307869-0 N429893-0 N668670-0

Dataset (cont.)

- news.tsv
  - **news_id**
  - **category**: news category.
  - **subcategory**: more fine-grained news category.
  - **title**: title of the news.
  - **abstract**: (possibly empty) abstract of the news.
  - **URL**

Dataset (cont.)

- news.tsv (cont.)
  - **title_entities**: special entities in news title that is linked with Wikidata. (Format: **json list**, each item is a linked entity)
  - **abstract_entities**: special entities in news abstract.

Dataset (cont.)

- entity_embedding.vec
  - For convenience, we provide **100-D** trained embeddings of special entities. (tab separated)
  - **You can also use your preferred embeddings.**

Kaggle

- Competition Link
- Create a team with your **group ID**, we use this information for grading. Use this link to find your group ID, your team name should be **group_<groupID>**.
- If you failed to do so under any circumstances, there will be **a penalty of 5 points** to your score, so be sure to use the correct team name.

Kaggle (cont.)

- Public leaderboard is calculated with 50% of the test set, private leaderboard is calculated with the other 50%, the final standings may be different.
- Therefore, **please DO NOT overtune your model to fit the public leaderboard, or you will suffer from overfitting**.

Kaggle (cont.)

- The scoring metric is **Area Under the Curve of ROC (AUC ROC)**. If you don't know what it is, here is a quick video for you.
- We have set a simple baseline and a strong baseline, beat them to get higher score.
- You can submit at most 10 times each day and choose 2 of the submissions to be scored for the private leaderboard, or will otherwise default to the best public scoring submissions.

Kaggle Submission Format

- Report user with their **index (column “id” in test_behaviors.tsv, not “user_id”)**, and the corresponding prediction to each news.
- There should be **46333 x 16** entries in your csv file, **with columns “id” and “p1 - p15”**.
- The order of the ids does not matter. Refer to sample_submission.csv for the correct format.

Kaggle Submission Format

- The “p1 - p15” columns should contain the probability predicted by your model, of the user clicking the 1st news, to the 15th news.
- Although we ask you to provide the “probability”, the value does not have to be in [0, 1], because the AUC metric still works without that rule.

Baseline Approach

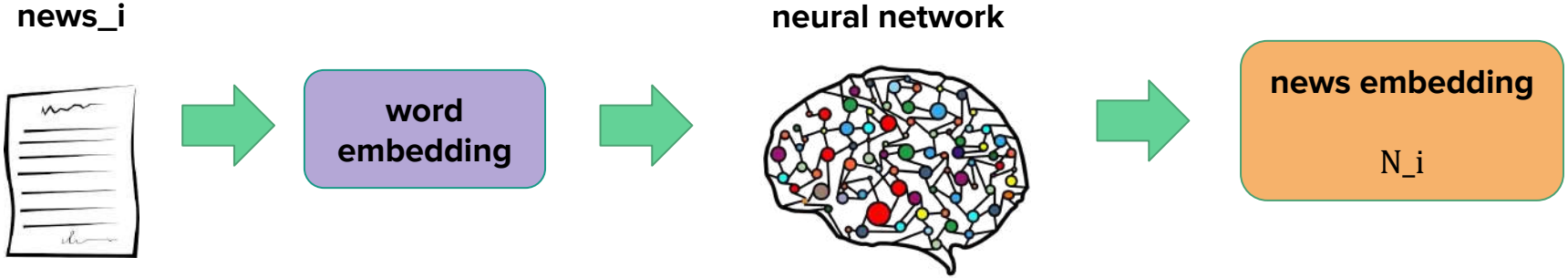
- **Disclaimer:** The following appoarch are just for your reference. You are not required to follow any of them. We encourage you to invent new ways to solve this problem.

Baseline Approach

- The amount of data is huge, and we need to predict 15 news for each user. The news itself is a short (possibly long) paragraph, we won't be able to fit them all in a single model. Therefore, try to find a way to **encode each news as a "news embedding"**. This will also help us comparing the similarity between each news.

Baseline Approach

- To build the news embedding, you can use BERT as the encoder, or train a neural network using your own preferred word embedding (e.g. GLoVe, flair).

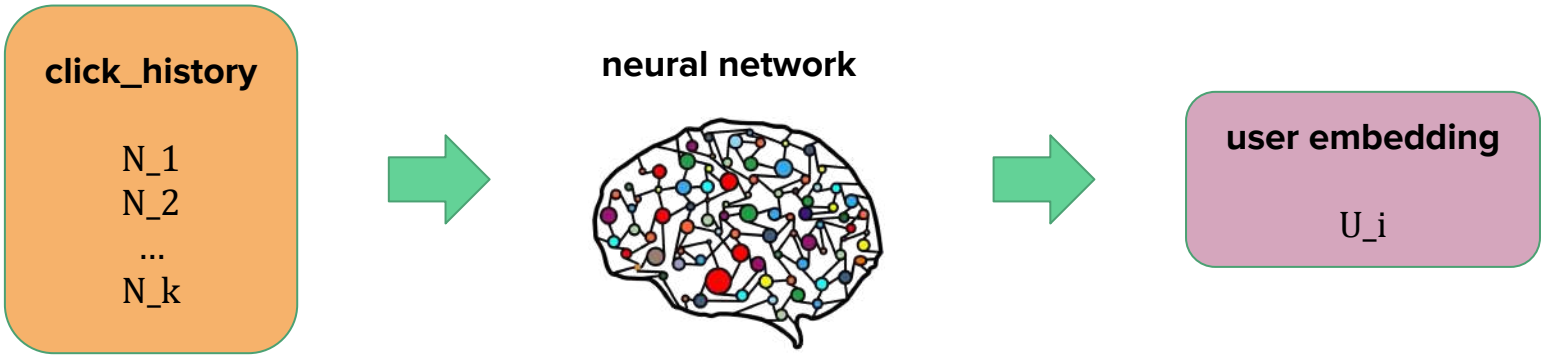


Baseline Approach

- For each user, we need to find news that are to their liking. To determine whether a user will like a news or not, we shall build **“user embedding”**, and then compare this embedding with the “news embedding” we just got.

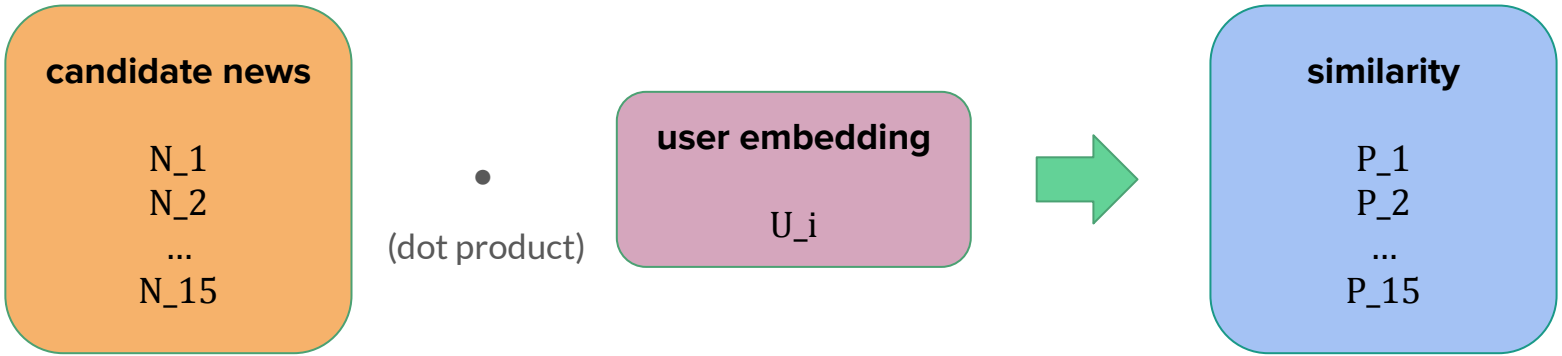
Baseline Approach

- One easy way to build a user embedding is to consider their click history, and use a model to train the embedding.



Baseline Approach

- Finally, to calculate the similarity, we can use **dot product** to compare the user embedding with each news embedding.



Tips

- There are a lot of additional information in the dataset, you can find a way to use them in your approach.
- LLMs may not help you predicting the news in this task, since the data is too big. However, since LLMs can receive huge inputs, you can leverage this to generate “enriched” description of user history, which may help your model predict better.

Report

Please answer the following questions, provide your thinking process as detailed as possible:

1. (10%) Describe how you solve this problem. Details include preprocessing, embeddings, model selection, hyparameters should be provided.

(There will be penalties to your score if you failed to do so)

Report (cont.)

2. (5%) Choose a **variable (e.g. different model, different approach)** **excluding hyperparameters** and compare their performance. **(You cannot change ONLY the hyperparameters)**

(5%) Explain what causes the difference of performance or why.

(For example, you were using BERT as the encoder, in Q2 you tried RoBERTa and found that using RoBERTa will yield better performance. Give a reason why this is the case)

Report (cont.)

3. (10%) Do some error analysis or case study. Is there anything worth mentioning while checking the mispredicted data? Share with us.

There is no “correct answer” to above questions, just do your best and answer them in detail.

Grading Policy

- **Kaggle (70%)**
  - 30% - public leaderboard, 70% - private leaderboard
  - Strong Simple Basic (AUC > 0.4)

    |---|---|

    Full point 70% 50%

- **Report (30%)**, points are attached at the start of each question.

E3 Submission

- Submit your source code and report to E3 before **6/21(Fri.) 23:59**, no late submissions will be accepted.
- Please submit your source code in **python source (.py)**. For jupyter notebooks, you can use the export function to obtain the executable script.
- **Do NOT put full code in your report.** Only short code snippets and pseudocode (for demonstration purposes) are allowed, and they should be properly formatted.

E3 Submission

Submission format:

- final_<group_id>.zip
  - **source code:** final_<group_id>.py or other library files (.py) you made
  - **report:** final_<group_id>.pdf

Contact

- If you have any questions about Final Project, please feel free to contact with TA by email:
  - Hank Chen (h7a4n1k.cs12@nycu.edu.tw)

- **Kindly format your email** so that I can reply it ASAP.
  - **Prefix your email title with [DM]**. (e.g. [DM] Questions about Final Project)
  - Attach your student ID if necessary.