Multilingual Natural Language Processing Social Advertising Recommendation System

Information Retrieval and Text Mining Final Project

Leonard Tsai	Po-Yen Chu	Shang-Ching Su	Zhe You	Kuan-Cheng Ku
B10705010	B10704031	B10704096	B09705018	B10705016
@ntu.edu.tw	@ntu.edu.tw	@ntu.edu.tw	@ntu.edu.tw	@ntu.edu.tw

1 Abstract

This paper develops a Social Advertising Recommendation System model, trained on diverse datasets comprising users treated as nodes. Each node is associated with historical purchase records, encompassing a comment, an item ID, a personal rating (in stars), a global rating (indicating helpfulness), and a timestamp. Additionally, the training data encompasses social relationships, represented as edges connecting these nodes, covered in the subsequent sections respectively.

The training process involves three key steps:

Trustworthiness Scoring First, this paper converted social relationships into trustworthiness scores between every pair of nodes. For instance, focusing on node i, we explore its relationship with another node j. Given that both i and j have multiple historical purchase records, we calculate a trustworthiness score for each of j's purchase records. This is achieved by leveraging the sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 model to determine the average cosine similarity between the comment of a specific j purchase record and all comments associated with i's purchase records.

Information Retrieval System Continuing with node i, we identify nodes with a degree less than k. Excluding node i, these nodes possess multiple purchase records, each containing an item ID, a personal rating (star), a global rating (indicating helpfulness of the comment), and a timestamp, along with the added dimension of trustworthiness. Utilizing these nodes, we construct an information retrieval system that takes into account trustworthiness, personal ratings, global ratings, and degree. We posit that the social network influences personal purchase behavior, suggesting that items in these nodes are likely to appear in node i's future purchase records.

Evaluation and Model Success Lastly, a time-based threshold is implemented, dividing 80% of purchase records as train X and the remainder as train y. Our focus

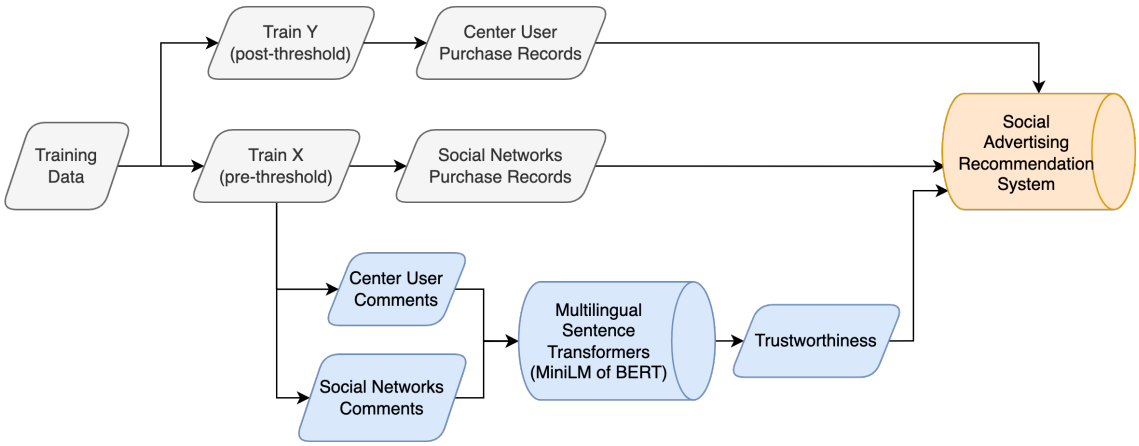

Figure 1: SARS Flow Chart

is on assessing whether the advertisements generated in steps 1 and 2 overlap with train Y. The success criterion is based on whether node i actually purchases the item advertised to itself by the model after the time threshold. The occurrence of such purchases indicates the model's efficacy.

2 Data Description

2.1 Dataset

The dataset is a real-world dataset that includes two main files: `reviews.txt` and `edges.txt` from LibraryThing. LibraryThing is a website that allows users to catalog their books, rate, and comment on them.

2.1.1 Data Format

`reviews.txt` The columns and their respective descriptions are as follows:

- `comment`: The comment that users publish for a specific item (book or other products). These comments are written in multiple languages.
- `work`: Equivalent to the item ID, which is a string of numbers representing a book or product.
- `user`: User ID, the ID created by users when registered.
- `stars`: The ratings given by users to the item they commented on.
- `nhelpful`: The number of users that view this comment as helpful.
- `time`: The real-world time (e.g., Jan 23, 2012) when the comment is published.
- `unixtime`: The Unix time when the comment is published.

`edges.txt` This document shows the connections between users without directions (two-way relationship). A partial view of the dataset is as follows:

```
Rodo anehan
Rodo sevilemar
Rodo dingsi
Rodo slash
RelaxedReader AnnRig
RelaxedReader bookbroke
...
```

The columns in `reviews.txt` provide a thorough understanding of each comment, enabling the construction of a system that leverages users' behavior and studies the

network by combining `edges.txt`.

2.1.2 Data Statistics

Here are some related statistics of the dataset:

- 1.7M reviews
- 73,882 users
- 337,561 items (calculated by work ID)
- 120,536 user relations (edges)
- File sizes
  - `reviews.txt`: 1.55 GB
  - `edges.txt`: 4.6 MB

The statistics, with such a large amount of data, imply that limited hardware may constrain the study of the entire graph.

2.2 Data Preprocessing

2.2.1 Network Graph Construction

This paper utilized Dijkstra's Shortest Path algorithm, a classic algorithm in computer science, to identify and analyze a subnetwork within a larger social network. The goal was to locate a specific subnetwork centered around a designated user, referred to as the 'central user'.

By implementing the algorithm, we calculate the shortest distances from this central user to every other user in the network. This process creates a map of connections detailing how far each user is from the central user in terms of network layers.

With these distances, we can then construct a subnetwork for any chosen depth 'k'. This k-deep subnetwork (**Figure 2**) includes all users who are within 'k' steps (or layers) from the central user. For instance, a 3-deep subnetwork will include users directly connected to the central user (1-deep), those connected to the 1-deep users (2-deep), and subsequently those connected to the 2-deep users (3-deep).

With the successful construction of k-deep subnetworks, we have developed a methodology to construct data feeds for model training. These feeds comprise the reviews of all users within the k-deep subnetwork, combined with an additional critical piece of information – the depth of each user from the central user. This approach is instrumental in providing a rich, contextual dataset for the model, enabling it to learn nuanced patterns based on user interactions and their proximity to the central user in the network.

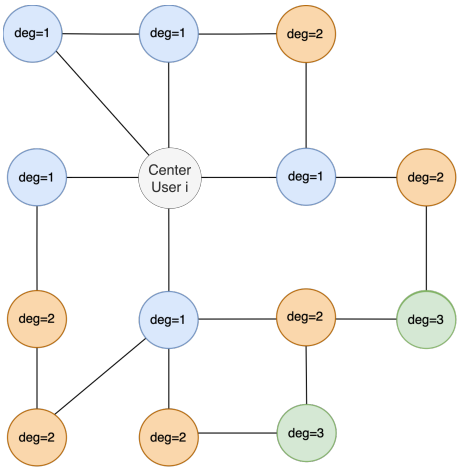

Figure 2: Subgraph when $k = 3$

2.2.2 Time Threshold Determination

This paper adopted a unique data splitting methodology for segregating our dataset into inputs and benchmarks. This methodology revolves around using a specific time threshold, determined by a specified factor, related to the timeline of the central user's review data. The primary objective of this approach was to ensure that a significant portion of the intersec-

tion of the central user's data and other users' data is present in the benchmarks, allowing for a more robust and realistic optimization of the system's performance. To be more precise, the time threshold is determined after ensuring $m\%$ of $\{center\_user \cap other\_users\}$ is in the benchmarks (**Figure 3**).

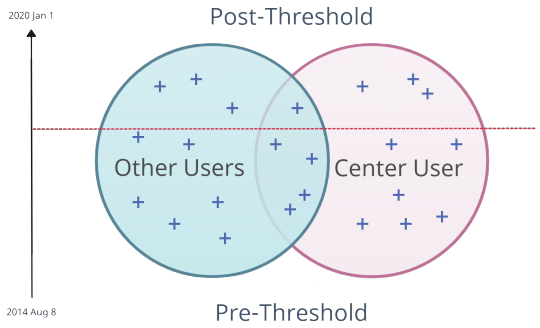

Figure 3: Inputs-benchmarks Splitting with 20% Time Threshold

The threshold is set such that the data before this point is used for training, and the data after this point is reserved to be viewed as benchmarks. This split is strategically chosen to mirror a realistic scenario where the model is trained on past data and validated on more recent data.

This approach creates a realistic scenario for model testing, simulating how the system would perform in real-world situations where it encounters new, unseen data. An instance in the dataset is shown in (**Figure 4**).

3 Trustworthiness Scoring

To train the model, it is necessary to append the data with the trustworthiness between comments from the central user and other users in the graph. Simply using the distance of friendship as the representation of trustworthiness is too monotonous and risky, and there needs to be other vital value to measure trustworthiness . This involves converting the variable-length data into fixed-length vectors to calculate cosine similarity, which is considered a measure of trustworthiness.

3.1 Transform

3.1.1 Sentence-transformers Model

The model used to modify data is a BERT-based model called paraphrase-multilingual-MiniLM-L12-v2. This model maps sentences to a 384-dimensional dense vector space and can be used for tasks such as clustering or semantic search. It allows for text compression without losing the semantic elements. The model's feature selection and multilingual properties are helpful for data compression.

3.1.2 Transforming

For the comments before the threshold, the text will be replace with the outcome generated from the text itself.

3.2 Cosine Similarity

3.2.1 Assumption

To establish trustworthiness, there is an assumption that a comment that closely aligns with the average comments of the central user holds more significance for the central user. The assumption's cornerstone is that if a comment is similar to that of the central user, there is a high probability that the central user will relate to it. This is because the two users have similar language habits (**Figure 5**).

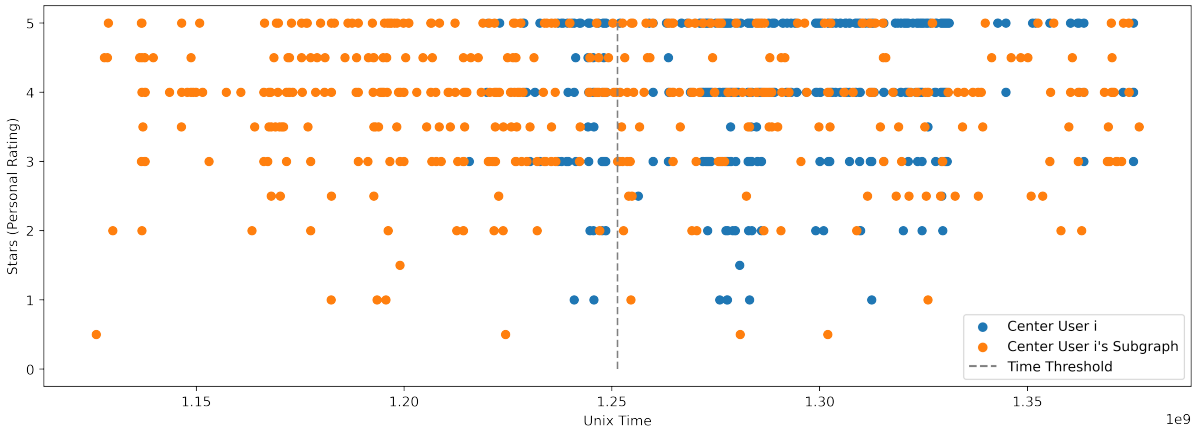

Figure 4: Purchase Records of Subgraph w.r.t User 'carterchristian1' having Common Goods Bought by the User

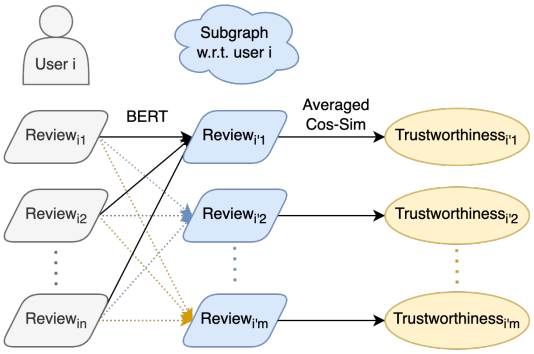

Figure 5: trustworthiness

3.2.2 Calculating

For each vector (extracted from the comment) from each user in the subgraph, excluding the central user, the cosine similarity will be compute with comments from the central user respectively. The trustworthiness of the comment will then be present as the arithmetic mean of the cosine similarities.

$$T_{i'k} = \frac{1}{n}\sum_{j=1}^{n} \text{sim}(\text{Review}_{ij}, \text{Review}_{i'k})$$

$T_{i'k}$: $\text{Trustworthiness}_{i'k}$

4 Model Construction & Optimization

The upcoming chapter will delve into our approach for evaluating and determining the promotion of specific items within our system. In addition, we will elaborate on our methodology for optimizing the scoring calculation, providing insights into the process of refining our recommendation system.

4.1 Ranking Items

To assess the scores of recommended products, our evaluation methodology operates on a subgraph basis, taking into consideration user-contributed factors within each subgraph.

4.1.1 Scoring Formula

The scoring factors encompass stars (user ratings), nhelpful (helpfulness index), trust (similarity of comments), degree (depth of subgraph). We have devised a formula, outlined as follows, incorporating diverse elements to comprehensively evaluate the scores of recommendations:

$$\sum_{k\in[1,m],\text{item}_{i'k}=item} D_{i',k}(S_{i',k}+H_{i',k}+T_{i',k})$$

$$D_{i',k}=\frac{\ln(\deg_{i'k})}{\ln(\deg_{i'}+1)}$$
$$S_{i',k}=\theta_1\frac{\text{star}_{i'k}-\overline{\text{star}_{i'}}}{\sigma_{\text{star}_{i'}}}$$
$$H_{i',k}=\theta_2\frac{\text{nhelpful}_{i'k}-\overline{\text{nhelpful}_{i'}}}{\sigma_{\text{nhelpful}_{i'}}}$$
$$T_{i',k}=\theta_3\frac{\text{trust}_{i'k}-\overline{\text{trust}_{i'}}}{\sigma_{\text{trust}_{i'}}}$$

deg : The depth of subgraph
star_i : The rating of user in subgraph i
nhelpful: The number of users that view this comment as a helpful comment
θ_i : θ_i is an adjustable parameter in the formula.

It is noteworthy that for the sake of consistency and comparability, each variable in the formula, including degree, star, nhelpful, and similarity, will undergo a standardization process. This standardization ensures that all variables are brought to a common scale, mitigating any potential biases arising from variations in measurement units.

4.1.2 Recommendations Driven by Score Evaluation

In practical implementation, our system will recommend products based on the calculated scores. The scores, determined by the evaluation formula, serve as the foundation for the recommendation mechanism. Higher scores indicate a stronger alignment with user preferences, and consequently, products with superior scores will be prioritized and recommended to users.

4.2 Formula Parameters Optimization

4.2.1 Criteria

To assess the effectiveness of the scoring formula and the achieved results, a maximization problem has been established. The primary objective is to identify the most suitable values for the parameters θs. The evaluation criteria are as follows:

$$\begin{aligned}\max_{\theta}\quad & \sum_{i\in N}\sum_{item\in Post_i}\text{score}(item,i)\\ \text{s.t.}\quad & 0\le\text{score}(\,\cdot\,)\le 1,\\ & \sum_{item\in set(item_{i'})}\text{score}(\,\cdot\,)\le c_0,\forall i\in N_{\text{Train}}\end{aligned}$$

N : The union of subgraphs
$Post_i$: Purchased items of subgraph i
$\text{score}(item,i)$: Score get from the predefined formula
c_0 : An assigned number to determine the minimum number of recommendations

4.2.2 Objective Function Value

The primary measure of success is the summation of score purchased by the central user of subgraph.

4.2.3 Constraints Satisfaction

Individual Score Constraint Each item's score is constrained to be less than or equal to 1. This prevents any individual item from receiving an excessively high score, promoting discrimination in the recommendation process.

Total Score Constraint The sum of all item scores is constrained to a predefined value. This constraint is introduced to ensure that the overall recommendation system maintains a balance, preventing the

algorithm from assigning uniformly high scores across all items.

4.2.4 Parameters Test

The table below displays the Theta values resulting from the optimization process across three selected subgraphs, chosen based on the criterion of a higher volume of data, as we aim to identify universal thetas reflecting prevalent trends within our recommendation system.

	θ_1	θ_2	θ_3
$c_0 = 5$	.0021600	.010840	.0003926
$c_0 = 6$	.0025922	.013006	.0004712
$c_0 = 7$	.0030243	.015173	.0005497
$c_0 = 8$	.0034563	.017341	.0006282
$c_0 = 9$	.0038884	.019509	.0007068
$c_0 = 10$	.0043204	.021676	.0007853

Table 1: Parameter Optimization

Notes: The three central users of subgraphs are "`slash`", "`Tay11`", "`Jeremiahstover`".

5 System Outcome & Performance Evaluation

To evaluate performance, we add a subgraph of center user "`SamSattler`" as the validation set, and rank by the pre-optimized system with $c_0 = 10$ in the previous section (which includes training subgraphs of center users "`slash`", "`Tay11`", "`Jeremiahstover`".

5.1 Ranking Items

By ranking items according to the assigned scores and examining the top 10 results based on user preferences, we identify one instance where a purchased item appears within this subset. This observation serves as an assurance that upon implementation of our recommendation system, at least 1 product is likely to capture the user's interest and drive a purchase at the first 10 recommendations.

It's crucial to highlight that the presence of top-scored products within the table doesn't discount the possibility that the center user may not have encountered all of the highest-rated items. Therefore, with the implementation of our recommendation system, the likelihood of selling products could be even higher, as users continue to explore and discover additional top-scored items.

itemID	**stars**	**score**	**rank**
4501506	0.0	0.06704	1.0
83185	0.0	0.06671	2.0
825285	0.0	0.06624	3.0
2753337	4.0	0.06324	4.0
7289	0.0	0.05854	5.0
862	0.0	0.05833	6.0
147503	0.0	0.05822	7.0
9979582	0.0	0.05806	8.0
83779	0.0	0.05749	9.0
27091	0.0	0.05664	10.0

Table 2: Top 10 highest scored recommendations

5.2 Unveiling Potential

It's crucial to highlight that the presence of top-scored products within the table doesn't discount the possibility that the center user may not have encountered all of the highest-rated items. Therefore, with the implementation of our recommendation system, the likelihood of selling products

itemID	stars	result	rank
2753337	4.0	0.06324	4.0
8499817	4.0	0.03823	29.0
9050484	3.0	0.03208	38.0
8300888	4.0	0.02379	75.0
10413252	4.0	0.01985	150.0

Table 3: Top 5 highest scored recommendations hit

could be even higher, as users continue to explore and discover additional top-scored items.

6 Conclusion & Recommendations

This paper proposes an effective approach for social platforms to promote products to users based on their network connections, introducing potentially attractive products that users might not have encountered before. The results presented highlight the system's potency, but certain predefined assumptions and limitations need consideration. Additionally, future recommendations are discussed.

6.1 Limitations & Assumptions

6.1.1 Memory & Time Complexity

While the memory required to process edges is modest, the memory demands for reviews are substantial, hindering the ability to train the entire graph. Despite this dataset being relatively small compared to current social media datasets, the time complexity of operations is noteworthy. Splitting with a time threshold incurs $O(n)$ complexity, where n is the number of reviews, and trustworthiness scoring requires $O(km)$, with k representing reviews of the center user and m representing those of other users. The stated time complexity excludes the transformers' complexity, making the system both memory and time-intensive.

Regarding flexibility, reviews publishing, relationship construction and new users necessitate system recalculations: reviews publishing forces the recalculation of trustworthiness scores in a subgraph, and relationship construction causes the recomputation of every related k-degree subgraph. Consequently, only large companies may effectively operate this recommendation system.

6.1.2 Data Limitations

Firstly, `edges.txt` is a graph without timestamps, assuming consistent relationships between users over the entire timespan. Additionally, although the dataset is split with a time threshold, users were not recommended by the system during this process. Thus, the actual accuracy could improve when the system is deployed.

6.2 Potential Improvements

6.2.1 Backtesting

Upon deployment, the system should include a backtesting function. Backtesting can dynamically readjust based on user behavior, leading to improved accuracy and potential flexibility enhancements.

6.2.2 Time Complexity and Flexibility Improvements

Addressing the issue of time complexity, future research could investigate the necessity of transformers for trustworthiness

scoring and explore methods to calculate the entire network to prevent overlapped relationship calculations when computing subgraphs independently.

The current ranking method is not conducive to adding new edges, reviews and nodes. Future research could explore an alternative system that accommodates real-time platforms, where edges and reviews are updated every second.

6.2.3 Imbalanced Effectiveness

The test results trained on three subgraphs for the parameters indicate that there is unequal influence among them. Specifically, the value of 'nhelpful' can be 30 times greater than the value of 'trustworthiness', which is unexpected. The initial goal was to use trustworthiness as a countervailing parameter to enhance uncertainty and flexibility. The significance of 'trustworthiness' is not apparent in this dataset, but might be another way around in other scenarios.

References

[1] McAuley, J., University of California, San Diego, *Recommender Systems and Personalization Datasets*. Accessed Dec 23, 2023, `https://cseweb.ucsd.edu/~jmcauley/datasets.html#social_data`.

[2] Cai, C., He, R., McAuley, J. (2017). *SPMC: Socially-Aware Personalized Markov Chains for Sparse Sequential Recommendation. arXiv preprint arXiv:1708.04497.*

[3] Zhao, T., McAuley, J., King, I. (2015). *Improving Latent Factor Models via Personalized Feature Projection for One-Class Recommendation.* In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), pp. 821–830.*

[4] Dijkstra, E. W. (1959). *A Note on Two Problems in Connection with Graphs. Numerische Mathematik, 1(1), p269–p271.*

[5] *paraphrase-multilingual-MiniLM-L12-v2.* Accessed Dec 23, 2023, `https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`.

[6] Reimers, N., Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks.*