ISSN 2007-9737

Using Text Embeddings and Graph Neural Networks for Personal Facts Classification

Konstantin Zaitsev*

HSE University, Faculty of Humanities, Moscow, Russia

kzaytsev@hse.ru

Abstract. In this study, we propose a framework for classifying personal facts in dialogue systems, utilizing a combination of lightweight text embeddings and Graph Neural Networks (GNNs). Due to the lack of existing labeled datasets, we annotated personal facts from the Multi-Session Chat (MSC) dataset using a large language model and verified these annotations manually. We categorize personal facts into Characteristics, Experiences, Routines or Habits, Goals or Plans, and Relationships. Our hypothesis is that semantically similar facts tend to share labels, enhancing classification accuracy. To test this, we construct a graph where nodes represent facts and edges reflect semantic similarity. Experimental results demonstrate that integrating GNNs with lightweight encoders consistently yields higher F1-scores than using encoders alone and rivals significantly larger models, indicating its efficacy in resource-limited environments. An ablation study further examines the roles of edge weighting and feature extraction in boosting classification performance. This work not only advances personal fact classification but also lays the groundwork for elevating the personalization of conversational agents.

Keywords. Personal facts classification, GNN, encoder models, text embeddings.

1 Introduction

In recent years, large language models (LLMs) have found application across a wide range of tasks, including in user communication interfaces. Within such interactions, LLMs can fulfill diverse roles—ranging from role-playing scenarios to providing intelligent assistance. In both cases, the LLM and the user may generate and exchange personal information, referred to as personal facts. These facts can be leveraged to enhance the quality and effectiveness of human–LLM interactions.

In particular, incorporating personal facts into the response generation process can contribute to a more personalized conversational experience. Recent studies demonstrate that integrating personal information into language models through prompts [19, 36, 15], embeddings [30, 2], or even graph-based representations [14, 22, 42] can improve user engagement and make conversations more natural and personalized.

Facts can be categorized into distinct types, including Characteristics, Experiences, Routines or Habits, Goals or Plans, and Relationships, with examples provided in Table 1. The classification of personal facts presents novel opportunities for enhancing personalized dialogue. Specifically, a classifier for personal facts can be employed in various applications, such as improving reranker models, the implementation of dialogue topic modeling, the summarization of dialogues through the distribution of personal fact labels, the development of recommender systems tailored to user preferences and characteristics, among other potential uses.

The integration of a personal facts classifier into LLMs necessitates consideration of its speed and quality performance, particularly in real-time applications. This requirement is especially actual to the classifier component, as it is responsible for categorizing newly extracted facts from dialogue and subsequently proposing related facts based on the extracted fact's label. Nevertheless, the

ISSN 2007-9737

Table 1. Labeled Personal Facts Samples

Facts	Labels
A lot of my family members are teachers.	Relationship, Experiences
I am afraid of snakes.	Characteristics
i live alone with my cats.	Experiences
Each morning I make an omelet with 6 eggs.	Routines or Habits
I love making apple pie and apple cider.	Characteristics, Routines or Habits
I want to buy a muscle car.	Goals or Plans

development of an optimal classifier presents challenges. A classifier with a small number of parameters may compromise the overall quality, whereas a classifier with a large number of parameters may achieve higher accuracy yet operate with reduced speed.

To tackle these issues, we hypothesize that semantically similar facts are likely to be assigned identical labels, suggesting that graph-based models are the most suitable framework for capturing such semantic relationships. By developing graph-based models and subsequently integrating them with a personal facts classifier, the accuracy of the classifier can be enhanced. This integration allows the classifier to gain insights from adjacent facts, thereby improving its ability to categorize facts with greater precision and consistency.

To validate the proposed hypothesis, it is necessary to perform a classification of personal facts according to specific thematic categories. To the best of our knowledge, there is no publicly available dataset specifically designed for our research. This lack of data necessitates the creation of an annotated dataset and the training of models on a limited amount of information.

In this paper, we present a detailed description of our proposed method for the classification of personal facts, outlining our approach to dataset construction and annotation. Furthermore, we evaluate the performance of our integrated classification model in comparison with baseline approaches.

Our contributions are as follows:

— **Development of a Personal Facts Classification Dataset:** Given the lack of existing datasets related to personal facts classification, we contribute a dataset that is annotated automatically and subsequently validated manually. This contribution not only facilitates the development of the classification framework but also serves as a foundation for subsequent research and analysis of the utilization of personal facts in dialogues.

— **A Framework for Personal Facts Classification:** We develop a framework that combines the strengths of text embeddings and GNNs. This framework effectively captures the semantic of personal facts through embeddings while also leveraging the graph structure of personal facts nodes via GNNs.

— **Analysis of Experimental Results of Combination Text Embeddings and GNN Models:** We conduct an ablation analysis of various methodologies for training classification models focused on personal facts. Moreover, we demonstrate the comparison of employing models with varied weight configurations to effectively extract features from text sources and construct relationships among nodes within a graph framework.

2 Related Work

Graph Neural Networks (GNNs) are a class of deep learning models designed to process graph structures. There are numerous GNN architectures, including Graph Convolutional Network (GCN) [17], Graph Attention Network (GAT) [31], and GraphSAGE [10]. These networks are employed in tasks where the graph structure serves as a critical source of information, such as predicting node [13, 12], link [21, 11, 33] and graph [41, 25] properties.

In the context of text classification, it is important to derive meaningful features from textual data. Traditional approaches, including one-hot encodings, bag-of-words (BOW) and TF-IDF, are commonly employed in classification tasks [24, 26, 3, 7]. Nevertheless, more sophisticated methodologies that employ neural networks demonstrate enhanced efficacy in recent years. In many cases, text embedding models are employed as feature extractors [28, 9, 16].

Text features, whether derived from TF-IDF or neural models, can be integrated as input features for GNNs. For example, a study [38] addresses node classification tasks by constructing a heterogeneous graph where nodes represent words and articles, and edges are weighted by TF-IDF scores and Pointwise Mutual Information (PMI). However, the embeddings in this approach are not fine-tuned during training.

Fine-tuning embeddings can be computationally expensive, which presents a challenge that the authors [23] address by training models in batches. They train a BERT encoder [4] and a GNN together for the node classification tasks. As a result, this combination shows the highest accuracy across approaches that fine-tune only text embeddings.

Personal facts can be classified based on their inherent characteristics. A paper [8] identifies five distinct fact classes, which were defined based on the analysis of existing literature. Another paper [18] proposes a similar categorization, further validating the consistency and relevance of fact classes across different research efforts.

3 Data

3.1 Dataset

There are a limited number of datasets that contain dialogues with annotated facts. Examples of such datasets include PersonaChat [40] and Multi-Session Chat (MSC) [35]. Both datasets consist of conversations between users, where each dialogue participant is associated with a set of personal facts, referred to as a persona. In contrast to PersonaChat, MSC is divided into multiple sessions, simulating real-life scenarios where conversations may be interrupted by time intervals. The MSC dataset is the most suitable for our purposes because it contains simple annotations in which a user's utterance can contain a personal fact.

Although personal facts can be represented in text form, if the classifier only works with text embeddings, such a representation may lack sufficient informativeness. The paper [8] proposes the use of personas in a knowledge graph representation (PeaCoK). In this graph, personas are nodes, and edges represent relations between them. Furthermore, they propose a classification of these relations based on works devoted to the analysis of human conversation.

Nevertheless, it is also feasible to categorize the nodes themselves, rather than merely the relations between them. In our approach, the graph of personal facts is presented in a slightly simplified manner compared to how it would be constructed as a knowledge graph. It thus follows that our graph is undirected. This structure allows the creation of edges between facts in such a way that the most semantically close facts will be neighbours. We follow the personal facts classification described in PeaCoK:

— **Characteristics** describe mental state, personal traits, preferences, moods.

— **Experiences** describe events that occurred once or in the past. It may relate to the current work, marital status, etc.

— **Routines or Habits** describe frequently performed activities or activities the person does on a regular basis.

— **Goals or Plans** describe a person's wishes or actions that they want to achieve in the future.

— **Relationship** describes interactions with the other people. This label may overlap with other labels in cases where the utterance contains a fact about another person. For example, the fact ”My sister wants to go to university” has the labels ”Relationship” and ”Goals or plans”.

Table 2. Dataset Statistics

Label	**Train**	**Test**
Experiences	966	233
Characteristics	779	190
Routines or Habits	286	71
Goals or Plans	285	65
Relationship	105	28
Overall	2421	587

As can be seen with the ”Relationship” label, each fact can have multiple labels. For example, a single fact can be related to both the ”Characteristics” and ”Routines or Habits” labels, as shown in the examples in the Table 1.

3.2 Personal Facts Labeling

Our study requires node classification, distinct from PeaCoK's relationship-based class system, prompting us to create a custom dataset with defined labels. Due to the high cost of manual annotation, we utilize an LLM for this process, following approaches highlighted in recent studies [29, 27, 1] demonstrating that LLMs can aid in generating high-quality annotations.

We employ an LLM[1] to annotate data from the MSC dataset, developing a detailed prompt with examples to ensure clarity. The output is structured in JSON format. Despite the LLM's utility, manual validation using Label Studio[2] is necessary, revealing errors in approximately 20% of cases.

To improve accuracy and efficiency, we implement an iterative, active learning-like approach: batches of 100 facts are manually validated, integrated into the dataset, followed by retraining and revalidating predictions. Ultimately, over 2,500 facts are validated and annotated, leveraging semantic repetition within the dataset. This process results in a dataset where each fact can have multiple labels, as detailed in Table 1 and summarized in Table 2.

[1] `https://huggingface.co/macadeliccc/WestLake-7B-v2-laser-truthy-dpo`

[2] `https://labelstud.io`

4 Approach

4.1 Graph Creation

TextGCN and BertGCN, which are based on text features and GNN, utilize heterogeneous graphs that connect words to articles. In contrast, our approach focuses on a homogeneous graph where edges are created between personal facts. Additionally, the existing studies propose weighting according to PMI or TF-IDF scores. However, in the context of personal facts classification, such an approach may not be as relevant.

The issue is that, rather than focusing on the statistics, it is necessary to consider the semantic similarity between the facts. This is because, in some cases, antonymous facts may have high TF-IDF scores but be semantically distant. For instance, the phrases ”I like ice cream” and ”I don't like ice cream” would have high TF-IDF scores but low scores for the semantic model.

A second challenge lies in the computational complexity of calculating similarities between nodes, which can be time-consuming due to the fully connected structure of the graph. However, it is not necessary to evaluate all possible node pairs, as many personal facts are clearly semantically dissimilar. To mitigate this issue, we propose an optimized graph construction approach.

Embeddings Extraction. To represent personal facts in a vectorized form, we first extract their text embeddings. For this purpose, we employ the BGE model developed by BAAI [34] in three different configurations—small, base, and large—which demonstrate strong performance on the MTEB leaderboard [6].

Neighbor Identification. Subsequently, we employ the k-nearest neighbours (k-NN) algorithm to ascertain the nearest neighbours for each personal fact. We select cosine similarity as the metric. It is noteworthy that the number of neighbours is a hyperparameter that can be adjusted. In our experiments, the optimal choice is seven neighbours.

Edge Weighting. For each group of neighboring nodes, we calculate pairwise similarity scores using both Natural Language Inference (NLI) and text embedding models. The inclusion

ISSN 2007-9737

of NLI aims to investigate whether it can enhance accuracy compared to embedding-based approaches, particularly in identifying antonymous facts, which embedding models may fail to capture effectively. In the NLI-based formulation, we generate all possible pairs of neighboring personal facts and compute entailment scores using the DeBERTa classifier [20]. Simultaneously, similarity scores are obtained using the BGE models, as described in the Embeddings Extraction section. The resulting scores are then utilized as edge weights in the graph structure.

Graph Construction. Finally, we construct a weighted graph where the nodes represent personal facts and the edges are weighted by the similarity scores obtained from the NLI model. We retain all edges regardless of weight, as we assume that even small edge weights provide valuable information for classification.

The resulting graph is a weighted homogeneous structure. The proposed method enables efficient graph construction while maintaining a high degree of semantic fidelity between personal facts. An example of the constructed graph is presented in Figure 1.

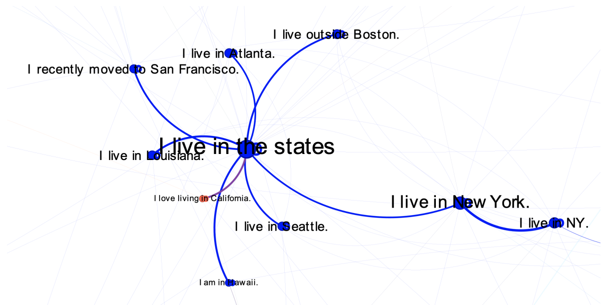

Fig. 1. An example of a subgraph. Here the blue color is the label "Experiences", the red color is the multilabel "Experiences" and "Characteristics"

4.2 Model Architectures

In this section, we describe various model architectures, encompassing linear baselines frequently utilized in machine learning, integrations of diverse text representation models with GNNs, and the fine-tuning of text embedding models absent of GNN integration. The objective of these architectures is to investigate the effects of incorporating graph structures into the process of personal fact classification.

4.2.1 Linear Baselines

As a baseline, we use features produced by TF-IDF, BOW, and pre-trained text embeddings. We use these features as input to a classifier based on logistic regression and k-NN classifier. We do not fine-tune the embeddings in this case. This baseline serves as a point of comparison for more complex models.

4.2.2 Text Features with GNN Training

In this setup, we use the same features as in the linear baselines, but we replace the logistic regression or k-NN classifier with a GNN. The final layer of the GNN is a linear projection that outputs label probabilities. Importantly, we do not fine-tune the embeddings in this model either. We train only the GNN. This allows us to evaluate the effect of training on the embedding layer.

4.2.3 Text Embeddings Fine-Tuning

The next step is to fine-tune the text embeddings for the classification task. We use a text embedding model and a final linear layer that predicts probabilities of personal facts labels. This model helps us understand the importance of including a GNN in addition to the fine-tuned embeddings.

4.2.4 Text Embeddings Fine-Tuning with GNN Training

In the final model, text embeddings are fine-tuned jointly with the training of the GNN. This joint optimization is hypothesized to enhance overall model performance. The rationale behind this approach lies in the task-specific adaptation of the text embeddings, coupled with the structural information provided by the GNN.

GNNs are typically trained on the entire dataset. While this approach poses no significant challenges for the GNN itself, fine-tuning text embeddings across all data samples can be

ISSN 2007-9737

computationally intensive. To address this issue, we adopt a batching strategy similar to that proposed in BertGCN. During each training epoch, text embeddings are computed and stored in a matrix, which is subsequently updated during the forward pass. It is important to emphasize that the matrix itself is not fine-tuned; instead, the fine-tuning is applied to the text embedding model.

The model is designed to classify text by leveraging both a pre-trained text encoder and a GNN. The encoder captures the semantic content of the text, while the GNN incorporates graph-based relational information. The output logits from both components are combined using a weighted sum to produce the final classification logits.

The encoder component uses a pre-trained model (e.g., BGE) to extract semantic features from the input text. The feature extraction can be represented as:

$$H = Encoder(X). \quad (1)$$

In this equation, H is the hidden representation obtained from the encoder for the input tokens X. The extracted features H are fed into a classifier head to produce the classification logits ($Z_{Encoder}$):

$$Z_{Encoder} = ClassifierHead(H). \quad (2)$$

The $ClassifierHead$ consists of a dense layer followed by a dropout layer and an output projection layer.

The GNN component takes the encoder features and additional graph-based information to compute another set of classification logits (Z_{GNN}). The GNN utilizes node features, edge index, and edge weight:

$$Z_{GNN} = GNN(H, edge_{index}, edge_{weight}). \quad (3)$$

The final logits (Z) are obtained by combining ($Z_{Encoder}$) and (Z_{GNN}) using a weighted sum, controlled by the hyperparameter (λ):

$$Z = \lambda \cdot Z_{GNN} + (1 - \lambda) \cdot Z_{Encoder}. \quad (4)$$

We set (λ) to 0.7, following the implementation inspired by BertGCN.

5 Experiments Setup

We conduct a series of experiments involving various approaches, model architectures, and parameter scales to evaluate the effectiveness of incorporating graph structures. The primary objective is to demonstrate that the integration of graph-based representations can enhance the performance of relatively simple models, even when trained on limited data and with smaller model sizes. Since a personal fact may correspond to multiple labels, the task is formulated as a multi-label classification problem.

For the GNN model, we employ GraphSAGE due to its demonstrated superior performance relative to alternative GNN models. The input sequence length for the text embedding model is fixed at 256 tokens, with a batch size of 32. To facilitate effective convergence, we assign distinct learning rates to different components of the model: 2e-4 for the text encoder and 2e-3 for the GNN. For text representation, we utilize BGE models of varying sizes (small, base, and large).

To comprehensively analyze the impact of incorporating GNNs, the dataset is partitioned into multiple train-test splits with varying ratios. For each split, experiments are conducted over 10 independent runs to ensure statistical reliability. All models are trained for 20 epochs, with the exception of those that do not involve fine-tuning of text embeddings; these are trained for 1000 epochs to compensate for their slower convergence. To mitigate the risk of overfitting, model selection is based on the highest F1 score achieved on the test set.

In order to evaluate the performance of the models, we select three metrics: F1, precision, and recall. These metrics are chosen due to the fact that the dataset presents an imbalanced distribution of labels. For each model, we identify the optimal threshold for the probabilities, which corresponds to the highest F1 score. Given that the task is that of a multi-label classification task, we choose Binary Cross-Entropy with Logits as the loss function.

ISSN 2007-9737

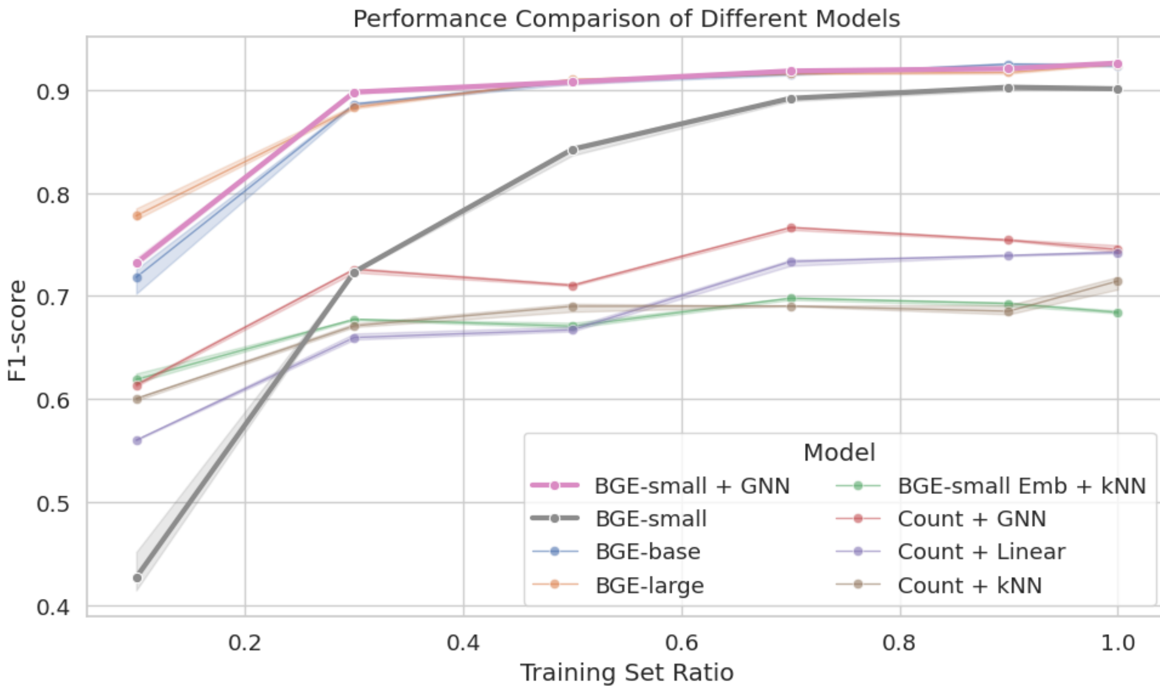

Fig. 2. Performance comparison of different models across varying training set ratios. The F1-score is used as the evaluation metric. ”BGE-small”, ”BGE-base”, and ”BGE-large” represent different sizes of the BGE model. ”+ GNN” indicates the integration of a GNN module into the encoder. ”Emb + kNN” shows model with kNN built on text embeddings, ”+ Linear” shows a linear baseline model

6 Results

This section presents the experimental evaluation of results of various text embedding model architectures and sizes, both with and without integration into GNNs. To further understand the influence of individual components, an ablation study is performed, examining the contributions of different feature extraction techniques and edge weighting strategies to the overall model performance.

6.1 Overall Performance

To begin, we assess the overall classification performance achieved by the proposed model configurations. Figure 2 presents the F1-scores obtained by each configuration across different training set ratios. As depicted, the pipelines incorporating GNNs (”BGE-small + GNN” and ”Count + GNN”) consistently outperform their non-GNN counterparts across nearly all training proportions. This consistent improvement indicates that the integration of relational information—captured through semantic similarity scores and represented within the graph structure—offers a substantial benefit to the classification performance.

Specifically, ”BGE-small + GNN” exhibits a significant improvement over the baseline ”BGE-small”, particularly when the training data is scarce (training set ratios below 0.4). This indicates that the GNN's ability to propagate information between semantically related nodes is particularly beneficial when the model has limited exposure to individual personal facts. The performance gap narrows as the training set size increases, but ”BGE-small + GNN” maintains a consistent, albeit smaller, advantage. ”Count + GNN” shows a boost

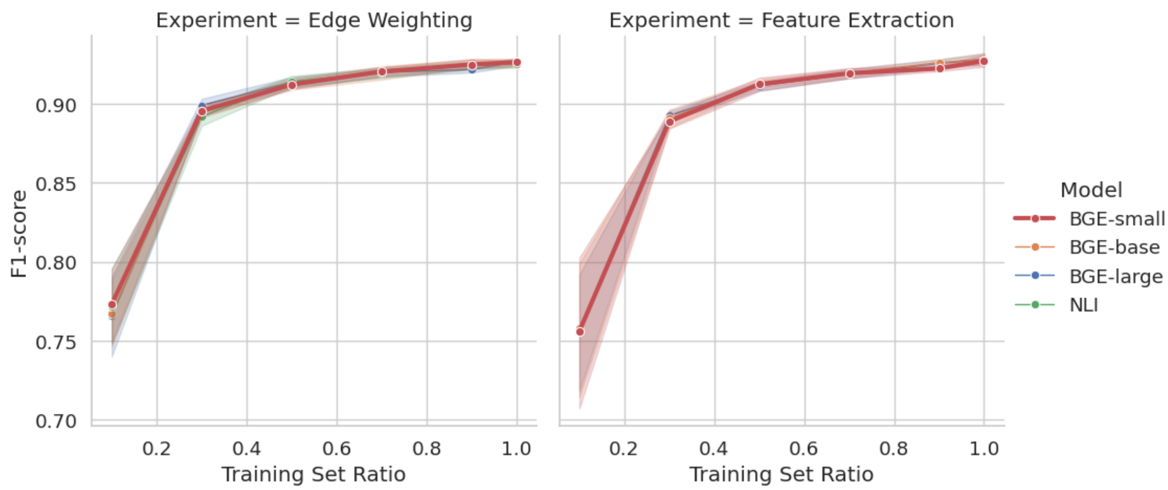

Fig. 3. Performance comparison of the text embedding pipeline and GNN in experiments with different models for edge weighting (left) and node feature extraction (right)

Table 3. Number of parameters for different models used in the experiments. The addition of a GNN to the BGE-small model results in a modest increase in the total parameter count

Model	# Parameters
BGE-small	33,361,925
BGE-small + GNN	33,365,770
BGE-base	109,486,085
BGE-large	335,147,013

in performance when compared to ”Count + kNN” and ”Count + Linear”.

The ”BGE-small Emb + kNN” model, which relies on k-NN applied to embeddings, demonstrates inferior performance compared to the ”BGE-small + GNN” configuration. Furthermore, even the linear baseline models outperform the ”BGE-small Emb + kNN” approach. These results underscore the effectiveness of the GNN's iterative message-passing mechanism, which enables more advanced and context-aware aggregation of information from neighboring nodes, surpassing the capabilities of the simpler k-NN-based method.

Across nearly all training set proportions, the ”BGE-small + GNN” configuration demonstrates performance comparable to that of larger BGE sizes (base and large). This observation suggests that integrating a GNN into the classification pipeline with a lightweight embedding model yields significant performance gains, on par with those achieved by substantially larger models. Notably, the overall size of the combined embedding model and GNN remains relatively small, due to the limited number of parameters in the GNN component, as is shown in Table 3. Consequently, this compact configuration achieves results comparable to larger models while maintaining a low parameter count. These findings not only support our hypothesis regarding the performance benefits of incorporating GNNs but also highlight the practical advantages of this approach for real-world applications where model size and efficiency are critical.

6.2 Impact of Edge Weights Computation Model

In the graph construction process, two critical components are the methods used for computing edge weights between nodes and the approaches for extracting features to represent graph nodes. To investigate the impact of model size on the formation of graph connections, we conduct a series of experiments. Specifically, we evaluate several embedding models, including the BGE model of different sizes (small, base, and large), as well as an NLI model. The inclusion of the NLI model is motivated by the assumption that it may be better suited for handling factual relationships involving negation compared to semantic embedding models.

Figure 3 presents a comparison of model performance across different training set proportions for embedding models of various sizes, as well as the NLI-based model. The results indicate that all models achieve comparable levels of performance. Moreover, the NLI model does not yield a noticeable improvement in overall pipeline accuracy and, in some cases, performs worse than the semantic embedding models. These findings suggest that our initial hypothesis regarding the superior performance of the NLI model—particularly in handling negations—does not hold in practice.

Given that models of different sizes exhibit comparable performance, it can be concluded that BGE-small is suitable for use in constructing graph edges. This eliminates the need to incorporate larger models into the pipeline, thereby optimizing the overall framework for personal fact classification in terms of computational efficiency and resource usage.

6.3 Impact of Feature Extraction Model

The next stage of our ablation study focused on examining the impact of embedding model size on the quality of feature extraction for graph nodes and, consequently, on the overall pipeline performance. As in the previous section, we employ BGE models of varying sizes for this set of experiments.

Figure 3 presents a comparison of model performance across different training set proportions. The results demonstrate that all models achieve similar performance levels at each training size ratio, indicating that the size of the embedding model used for feature extraction does not significantly impact the overall pipeline's effectiveness.

This observation is consistent with earlier findings related to edge construction. Accordingly, a lightweight model such as BGE-small is sufficient for use across all stages of the pipeline. These results confirm that BGE-small can be effectively employed throughout the entire process, including training and graph construction, while maintaining performance comparable to that of larger models.

7 Future Work

The proposed framework can be integrated into the Retrieval Augmented Generation (RAG) architecture, as several studies have demonstrated the inclusion of RAG approaches within personalized dialogue systems [32, 37, 39]. Motivated by the RAG paradigm, a reranker model could be employed to propose relevant personal facts based on their thematic labels and the conversational context. For example, when a personal fact extracted from a user utterance is categorized under "Characteristics," a tailored list of personal facts associated with this label could be constructed for the dialogue model. By utilizing this constructed list, the dialogue model is guided to acknowledge specific labeled personal facts in the generation of personalized responses. This methodology is particularly advantageous in scenarios necessitating the reduction of context size usage, as it avoids embedding the entirety of personal facts in the prompt, thereby allowing for a concentrated focus on particular facts to enhance personalization.

Furthermore, the constructed graph of labeled personal facts could be efficiently maintained in various vector databases such as Elasticsearch[3] or faiss [5]. This configuration ensures that the incorporation of newly extracted personal facts remains computationally efficient, thereby preserving the classification accuracy.

8 Conclusion

In this paper, we presented a framework for classifying personal facts into thematic categories by integrating lightweight text embeddings with Graph Neural Networks (GNNs). Our experiments revealed that a model pipeline utilizing a compact encoder, such as BGE-small, combined with a GNN, delivers classification performance on

[3] `https://github.com/elastic/elasticsearch`

par with significantly larger models. These findings confirm that the incorporation of relational information via graph structures enhances the capabilities of resource-efficient models, rendering them suitable for deployment in scenarios where computational resources are constrained. This work not only validates the potential of using GNNs for personal fact classification but also highlights a promising approach for advancing the personalization in conversational agents by effectively leveraging semantic relationships.

References

1. **Aguda, T. D., Siddagangappa, S., Kochkina, E., Kaur, S., Wang, D., Smiley, C. (2024).** Large language models as financial data annotators: A study on effectiveness and efficiency. **Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., Xue, N.**, editors, Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, pp. 10124–10145.

2. **Cheng, J., Sabour, S., Sun, H., Chen, Z., Huang, M. (2023).** PAL: Persona-augmented emotional support conversation generation. **Rogers, A., Boyd-Graber, J., Okazaki, N.**, editors, Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, pp. 535–554. DOI: `10.18653/v1/2023.findings-acl.34`.

3. **Das, B., Chakraborty, S. (2018).** An improved text sentiment classification model using tf-idf and next word negation.

4. **Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for language understanding. **Burstein, J., Doran, C., Solorio, T.**, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. DOI: `10.18653/v1/N19-1423`.

5. **Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., Jégou, H. (2024).** The faiss library. .

6. **Enevoldsen, K., Chung, I., Kerboua, I., Kardos, M., Mathur, A., Stap, D., Gala, J., Siblini, W., Krzemiński, D., Winata, G. I., Sturua, S., Utpala, S., Ciancone, M., Schaeffer, M., Misra, D., Dhakal, S., Rystrøm, J., Solomatin, R., Çağatan, Ö. V., Kundu, A., Bernstorff, M., Xiao, S., Sukhlecha, A., Pahwa, B., Poświata, R., GV, K. K., Ashraf, S., Auras, D., Plüster, B., Harries, J. P., Magne, L., Mohr, I., Zhu, D., Gisserot-Boukhlef, H., Aarsen, T., Kostkan, J., Wojtasik, K., Lee, T., Suppa, M., Zhang, C., Rocca, R., Hamdy, M., Michail, A., Yang, J., Faysse, M., Vatolin, A., Thakur, N., Dey, M., Vasani, D., Chitale, P. A., Tedeschi, S., Tai, N., Snegirev, A., Hendriksen, M., Günther, M., Xia, M., Shi, W., Lù, X. H., Clive, J., K, G., Anna, M., Wehrli, S., Tikhonova, M., Panchal, H. S., Abramov, A., Ostendorff, M., Liu, Z., Clematide, S., Miranda, L. J. V., Fenogenova, A., Song, G., Safi, R. B., Li, W.-D., Borghini, A., Cassano, F., Hansen, L., Hooker, S., Xiao, C., Adlakha, V., Weller, O., Reddy, S., Muennighoff, N. (2025).** MMTEB: Massive multilingual text embedding benchmark. The Thirteenth International Conference on Learning Representations.

7. **Galke, L., Scherp, A. (2022).** Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide MLP. **Muresan, S., Nakov, P., Villavicencio, A.**, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, pp. 4038–4051. DOI: `10.18653/v1/2022.acl-long.279`.

8. **Gao, S., Borges, B., Oh, S., Bayazit, D., Kanno, S., Wakaki, H., Mitsufuji, Y., Bosselut, A. (2023).** PeaCoK: Persona commonsense knowledge for consistent and engaging narratives. **Rogers, A., Boyd-Graber, J., Okazaki, N.**, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, pp. 6569–6591. DOI: `10.18653/v1/2023.acl-long.362`.

9. **Garrido-Merchan, E. C., Gozalo-Brizuela, R., Gonzalez-Carvajal, S. (2023).** Comparing bert against traditional machine learning models in text classification. Journal of Computational and Cognitive Engineering, Vol. 2, No. 4, pp. 352–356. DOI: `10.47852/bonviewjcce3202838`.

10. **Hamilton, W. L., Ying, R., Leskovec, J. (2017).** Inductive representation learning on large graphs. CoRR, Vol. abs/1706.02216.

11. **He, J., Jia, L., Wang, L., Li, X., Xu, X. (2023).** Mocosa: Momentum contrast for knowledge graph completion with structure-augmented pre-trained language models.

12. **Huang, Q., He, H., Singh, A., Lim, S.-N., Benson, A. (2021).** Combining label propagation and simple models out-performs graph neural networks. International Conference on Learning Representations.

13. **Izadi, M. R., Fang, Y., Stevenson, R., Lin, L. (2020).** Optimization of graph neural networks with natural gradient descent. 2020 IEEE International Conference on Big Data (Big Data), pp. 171–179. DOI: `10.1109/BigData50022.2020.9378063`.

14. **Ju, D., Feng, S., Lv, P., Wang, D., Zhang, Y. (2022).** Learning to improve persona consistency in multi-party dialogue generation via text knowledge enhancement. **Calzolari, N., Huang, C.-R., Kim, H., Pustejovsky, J., Wanner, L., Choi, K.-S., Ryu, P.-M., Chen, H.-H., Donatelli, L., Ji, H., Kurohashi, S., Paggio, P., Xue, N., Kim, S., Hahm, Y., He, Z., Lee, T. K., Santus, E., Bond, F., Na, S.-H.**, editors, Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, pp. 298–309.

15. **Kasahara, T., Kawahara, D., Tung, N., Li, S., Shinzato, K., Sato, T. (2022).** Building a personalized dialogue system with prompt-tuning. **Ippolito, D., Li, L. H., Pacheco, M. L., Chen, D., Xue, N.**, editors, Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, pp. 96–105. DOI: `10.18653/v1/2022.naacl-srw.13`.

16. **Kementchedjhieva, Y., Chalkidis, I. (2023).** An exploration of encoder-decoder approaches to multi-label classification for legal and biomedical text.

17. **Kipf, T. N., Welling, M. (2016).** Semi-supervised classification with graph convolutional networks. CoRR, Vol. abs/1609.02907.

18. **Kumar, S., Gupta, R., Akhtar, M. S., Chakraborty, T. (2024).** Adding SPICE to life: Speaker profiling in multiparty conversations. **Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., Xue, N.**, editors, Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, pp. 415–425.

19. **Kwon, D., Lee, S., Kim, K. H., Lee, S., Kim, T., Davis, E. (2023).** What, when, and how to ground: Designing user persona-aware conversational agents for engaging dialogue. **Sitaram, S., Beigman Klebanov, B., Williams, J. D.**, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track), Association for Computational Linguistics, Toronto, Canada, pp. 707–719. DOI: `10.18653/v1/2023.acl-industry.68`.

20. **Laurer, M., van Atteveldt, W., Casas, A., Welbers, K. (2024).** Less annotating, more

classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli. Political Analysis, Vol. 32, No. 1, pp. 84–100. DOI: 10.1017/pan.2023.20.

21. **Li, H., Yu, B., Wei, Y., Wang, K., Xu, R. Y. D., Wang, B. (2024).** Kermit: Knowledge graph completion of enhanced relation modeling with inverse transformation.

22. **Li, S., Namazifar, M., Jin, D., Bansal, M., Ji, H., Liu, Y., Hakkani-Tur, D. (2022).** Enhancing knowledge selection for grounded dialogues via document semantic graphs. **Carpuat, M., de Marneffe, M.-C., Meza Ruiz, I. V.**, editors, Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, pp. 2810–2823. DOI: 10.18653/v1/2022.naacl-main.202.

23. **Lin, Y., Meng, Y., Sun, X., Han, Q., Kuang, K., Li, J., Wu, F. (2021).** BertGCN: Transductive text classification by combining GNN and BERT. **Zong, C., Xia, F., Li, W., Navigli, R.**, editors, Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, pp. 1456–1462. DOI: 10.18653/v1/2021.findings-acl.126.

24. **Lin, Y.-C., Chen, S.-A., Liu, J.-J., Lin, C.-J. (2023).** Linear classifier: An often-forgotten baseline for text classification. **Rogers, A., Boyd-Graber, J., Okazaki, N.**, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Toronto, Canada, pp. 1876–1888. DOI: 10.18653/v1/2023.acl-short.160.

25. **Nouranizadeh, A., Matinkia, M., Rahmati, M., Safabakhsh, R. (2021).** Maximum entropy weighted independent set pooling for graph neural networks. CoRR, Vol. abs/2107.01410.

26. **Qader, W. A., Ameen, M. M., Ahmed, B. I. (2019).** An overview of bag of words;importance, implementation, applications, and challenges. 2019 International Engineering Conference (IEC), pp. 200–204. DOI: 10.1109/IEC47844.2019.8950616.

27. **Rouzegar, H., Makrehchi, M. (2024).** Enhancing text classification through LLM-driven active learning and human annotation. **Henning, S., Stede, M.**, editors, Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII), Association for Computational Linguistics, St. Julians, Malta, pp. 98–111.

28. **Sun, C., Qiu, X., Xu, Y., Huang, X. (2020).** How to fine-tune bert for text classification?

29. **Tan, Z., Li, D., Wang, S., Beigi, A., Jiang, B., Bhattacharjee, A., Karami, M., Li, J., Cheng, L., Liu, H. (2024).** Large language models for data annotation: A survey.

30. **Tang, Y., Wang, B., Fang, M., Zhao, D., Huang, K., He, R., Hou, Y. (2023).** Enhancing personalized dialogue generation with contrastive latent variables: Combining sparse and dense persona. **Rogers, A., Boyd-Graber, J., Okazaki, N.**, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, pp. 5456–5468. DOI: 10.18653/v1/2023.acl-long.299.

31. **Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y. (2018).** Graph attention networks.

32. **Wang, H., Huang, W., Deng, Y., Wang, R., Wang, Z., Wang, Y., Mi, F., Pan, J. Z., Wong, K.-F. (2024).** Unims-rag: A unified multi-source retrieval-augmented generation for personalized dialogue systems.

33. **Wang, L., Zhao, W., Wei, Z., Liu, J. (2022).** SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. **Muresan, S., Nakov, P., Villavicencio, A.**, editors, Proceedings of the 60th Annual Meeting of the Association for Computational

ISSN 2007-9737

Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, pp. 4281–4294. DOI: 10.18653/v1/2022.acl-long.295.

34. **Xiao, S., Liu, Z., Zhang, P., Muennighoff, N. (2023).** C-pack: Packaged resources to advance general chinese embedding.

35. **Xu, J., Szlam, A., Weston, J. (2022).** Beyond goldfish memory: Long-term open-domain conversation. **Muresan, S., Nakov, P., Villavicencio, A.**, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, pp. 5180–5197. DOI: 10.18653/v1/2022.acl-long.356.

36. **Xu, X., Lei, Z., Wu, W., Niu, Z.-Y., Wu, H., Wang, H. (2023).** Towards zero-shot persona dialogue generation with in-context learning. **Rogers, A., Boyd-Graber, J., Okazaki, N.**, editors, Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, pp. 1387–1398. DOI: 10.18653/v1/2023.findings-acl.90.

37. **Yang, Y., Xu, C., Guo, J., Feng, T., Ruan, C. (2025).** Improving the rag-based personalized discharge care system by introducing the memory mechanism. 2025 IEEE 17th International Conference on Computer Research and Development (ICCRD), pp. 316–322. DOI: 10.1109/ICCRD64588.2025.10963086.

38. **Yao, L., Mao, C., Luo, Y. (2018).** Graph convolutional networks for text classification.

39. **Zerhoudi, S., Granitzer, M. (2024).** Personarag: Enhancing retrieval-augmented generation systems with user-centric agents.

40. **Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J. (2018).** Personalizing dialogue agents: I have a dog, do you have pets too? **Gurevych, I., Miyao, Y.**, editors, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, pp. 2204–2213. DOI: 10.18653/v1/P18-1205.

41. **Zhang, Z., Bu, J., Ester, M., Zhang, J., Li, Z., Yao, C., Dai, H., Yu, Z., Wang, C. (2023).** Hierarchical multi-view graph pooling with structure learning. IEEE Transactions on Knowledge and Data Engineering, Vol. 35, No. 1, pp. 545–559. DOI: 10.1109/TKDE.2021.3090664.

42. **Zhu, L., Li, W., Mao, R., Pandelea, V., Cambria, E. (2023).** PAED: Zero-shot persona attribute extraction in dialogues. **Rogers, A., Boyd-Graber, J., Okazaki, N.**, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, pp. 9771–9787. DOI: 10.18653/v1/2023.acl-long.544.

Article received on 08/06/2025; accepted on 30/09/2025.
**Corresponding author is Konstantin Zaitsev.*