RESEARCH-ARTICLE

CompanyKG: A Large-Scale Heterogeneous Graph for Company Similarity Quantification

LELE CAO, EQT AB, Stockholm, Sweden

VILHELM VON EHRENHEIM, EQT AB, Stockholm, Sweden

MARK GRANROTH-WILDING, Silo AI Sweden AB, Stockholm, Stockholms, Sweden

RICHARD ANSELMO STAHL, EQT AB, Stockholm, Sweden

ANDREW MCCORNACK, EQT AB, Stockholm, Sweden

ARMIN CATOVIC, EQT AB, Stockholm, Sweden

View all

Open Access Support provided by:

EQT AB

Silo AI Sweden AB

Published: 24 August 2024

Citation in BibTeX format

KDD '24: The 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining
August 25 - 29, 2024
Barcelona, Spain

Conference Sponsors:
SIGMOD
SIGKDD

CompanyKG: A Large-Scale Heterogeneous Graph for Company Similarity Quantification

Lele Cao*
Motherbrain, EQT Group
Stockholm, Sweden
caolele@gmail.com
lele.cao@eqtpartners.com

Vilhelm von Ehrenheim
Motherbrain, EQT Group
QA.tech
Stockholm, Sweden
vonehrenheim@gmail.com

Mark Granroth-Wilding
Motherbrain, EQT Group
Stockholm, Sweden
Silo AI, Helsinki, Finland
mark.granroth-wilding@silo.ai

Richard Anselmo Stahl
richard.stahl@eqtpartners.com
Motherbrain, EQT Group
Stockholm, Sweden

Andrew McCornack
Armin Catovic
<first_name>.<last_name>@eqtpartners.com
Motherbrain, EQT Group
Stockholm, Sweden

Dhiana Deva Cavalcanti Rocha
dhiana.deva@eqtpartners.com
Motherbrain, EQT Group
Stockholm, Sweden

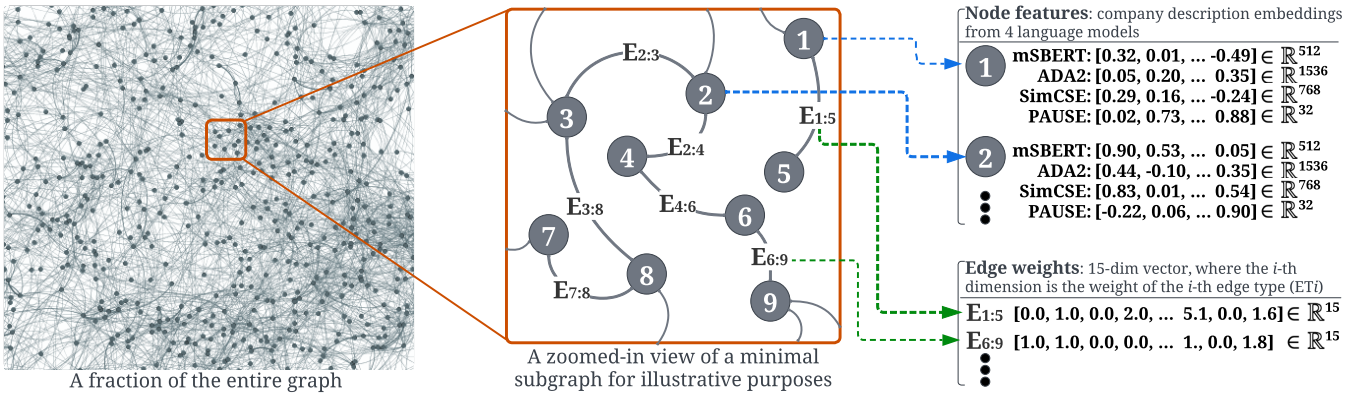


Figure 1: A subgraph (center) of the heterogeneous undirected CompanyKG graph (left), where the numbered nodes represent distinct companies and the edges signify multi-dimensional inter-company relations. Node features and edge weights are exemplified on the right.

Abstract

This paper presents CompanyKG (version 2), a large-scale heterogeneous graph developed for fine-grained company similarity quantification and relationship prediction, crucial for applications in the investment industry such as market mapping, competitor analysis, and mergers and acquisitions. CompanyKG comprises 1.17 million companies represented as graph nodes, enriched with company description embeddings, and 51.06 million weighted edges denoting 15 distinct inter-company relations. To facilitate a thorough evaluation of methods for company similarity quantification and relationship prediction, we have created four annotated evaluation tasks: similarity prediction, competitor retrieval, similarity ranking, and edge prediction. We offer extensive benchmarking results for 11 reproducible predictive methods, categorized into three groups: node-only, edge-only, and node+edge. To our knowledge, CompanyKG is the first large-scale heterogeneous graph dataset derived from a real-world investment platform, specifically tailored for quantifying inter-company similarity and relationships.

CCS Concepts

• **Computing methodologies** → **Knowledge representation and reasoning**; *Unsupervised learning*; Neural networks; *Natural language processing*; • **Mathematics of computing** → *Graph algorithms*; • **Information systems** → **Retrieval models and ranking**; *Enterprise applications.*

Keywords

knowledge graph, company similarity quantification, edge prediction, benchmark, graph neural network, investment, private equity.

*Send correspondence of CompanyKG version 2 (dataset: https://zenodo.org/records/11391315, source code and tutorial: https://github.com/llcresearch/CompanyKG2) to Lele Cao. This is a major extension from the previous version (V1) [8].

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671515

ACM Reference Format:
Lele Cao, Vilhelm von Ehrenheim, Mark Granroth-Wilding, Richard Anselmo Stahl, Andrew McCornack, Armin Catovic, and Dhiana Deva Cavalcanti Rocha. 2024. CompanyKG: A Large-Scale Heterogeneous Graph for Company Similarity Quantification. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671515

1 Introduction

In the investment industry, it is often essential to identify similar companies for purposes such as market mapping, competitor analysis[1] and mergers and acquisitions (M&A)[2]. Identifying similar and related companies in these scenarios is crucial, as they inform investment decisions and discover potential synergies. Therefore, the accurate quantification of inter-company similarity (a.k.a., company similarity quantification) is the key to successfully executing such tasks. Formally, company similarity quantification refers to the systematic process of measuring and expressing the degree of likeness or resemblance between two companies with a focus on various "traits", typically represented through numerical values. While a universally accepted set of "traits" does not exist, investment researchers and practitioners commonly rely on attributes, characteristics or relationships that reflect competitive landscape, industry sector, M&A transactions, people's affiliation, news/event engagement, product positioning, and more.

In data-driven investment activities, company similarity quantification often forms a central component in a company expansion system, which aims to extend a given set of companies by incorporating the similarity traits exemplified previously[3]. Such a system aids in competitor analysis, where investment professionals begin with a seed company x_0 of interest (a.k.a., query company, represented by a company ID), aiming to identify a concise list of direct competitors to x_0. The identified competitor companies serve as benchmarks for evaluating the investment potential of x_0, influencing the final investment decisions. Until now, implementing such a company expansion system remains a challenge, because (1) the crucial business impact necessitates an exceptionally high production-ready standard of expansion recall and precision; and (2) the underlying data has expanded to an unprecedented scale and complexity, particularly regarding inter-company relations.

In recent years, Transformer-based Language Models (LMs) [6, 10, 11, 15, 37] have become the preferred method for encoding textual company descriptions into vector-space embeddings. Companies that are similar to the seed companies can be searched in the embedding space using distance metrics like cosine similarity. Rapid advancements in Large LMs (LLMs), such as GPT-3/4 [5, 32], LLaMA [43] and Gemini-1.5 [36], have significantly enhanced the performance of general-purpose conversational models. Models such as ChatGPT [33] can, for example, be employed to answer questions related to similar company discovery and quantification in an N-shot prompting paradigm.

A graph is a natural choice for representing and learning diverse company relations due to its ability to model complex relationships between a large number of entities. By representing companies as nodes and their relationships as edges, we can form a knowledge graph (KG)[4]. A KG allows us to efficiently capture and analyze the network structure of the business landscape. Moreover, KG-based approaches allow us to leverage powerful tools from network science, graph theory, and graph-based machine learning, such as Graph Neural Networks (GNNs) [16, 18, 21, 41, 45, 56], to extract insights and patterns to facilitate similar company analysis that goes beyond the relations encoded in the graph. While there exist various company datasets (mostly commercial/proprietary and non-relational) and graph datasets for other domains (mostly for single link/node/graph-level predictions), there are to our knowledge no datasets and benchmarks that target learning over a large-scale KG expressing rich pairwise company relations.

EQT Motherbrain[5] is a general-purpose and proprietary investment platform developed by a team of researchers, engineers and designers. The platform integrates over 50 data sources[6] about companies, providing EQT's investment professionals with a comprehensive and up-to-date view of the individual companies and the entire market. Meanwhile, valuable insights are collected from the interaction between our professionals and the platform, which often implies inter-company relations of various kinds. Based on this unique edge, we construct and publish a large-scale heterogeneous graph dataset to benchmark methods for similar company quantification. Our main contributions include:

- We introduce a real-world KG dataset – CompanyKG – for similar company quantification and relation prediction. Specifically, 1.17 million companies are represented as nodes enriched with company description embeddings; and 15 distinct inter-company relations result in 51.06 million weighted edges.
- To facilitate a comprehensive assessment of methods learned on CompanyKG, we design and collate four evaluation tasks with annotated datasets: Similarity Prediction (SP), Competitor Retrieval (CR), Similarity Ranking (SR) and Edge Prediction (EP).
- We provide comprehensive benchmarking results (with source code for reproduction) of node-only (embedding proximity and N-shot prompting), edge-only (shortest path and direct neighbors) and node+edge (self-supervised graph learning) methods.

2 Related work

Modeling company similarity typically employs a multidimensional approach, encompassing facets such as industry classifications [19],

[1] Market mapping analyzes the market landscape of a product or service, while competitor analysis focuses on the competitive landscape of a concrete company [51]. They both start with identifying a set of similar companies, yet the former often expects a more coarse and inclusive set of similar companies than the latter.

[2] M&A refers to the process of combining two or more companies to achieve strategic and financial objectives, such as enhanced market share and profitability [40]. Investment professionals typically work with a list of M&A candidate companies and examine them based on the sequence of their projected business compatibility.

[3] In domains beyond investment, related research such as [1, 28] focuses on expanding query entities to a more extensive and relevant set, akin to our work. We emphasize the difference to a company suggestion system which concentrates on proactively recommending entities based on user behavior and preferences. However, company expansion and suggestion are both concepts in the context of KG and recommender systems with slightly different processes and implication.

[4] A knowledge graph is a specific type of graph where nodes represent real-world entities/concepts and edges denote relations between the entities.

[5] EQT: https://eqtgroup.com; Motherbrain: https://eqtgroup.com/motherbrain.

[6] The complete list of data sources adopted by EQT Motherbrain can not be revealed due to legal and compliance concerns. However, we provide descriptions of the nature of the data sources from which CompanyKG is built, which should suffice for others with access to similar data sources to build a comparable KG for practical applications. We refer readers to [9] for review of common sources from a Venture Capital's view.

financial flexibility [20], and shared directorship [24]. Commonly used data sources in the investment domain, such as Diffbot, Pitchbook, Crunchbase, CB Insights and S&P Capital IQ[7], primarily offer predicted similar companies without revealing their methodologies. Investment practitioners have increasingly relied on LMs and KGs for scalable company similarity quantification. Our benchmark tasks and methods focus primarily on graph-based company similarity measurement, but a company KG of this sort could be used for other tasks like [23, 28]. We include non-KG-based approaches (embedding proximity and LLM prompting) for comparison, while also acknowledging other methods like document co-occurrence [13] or KG fusion with domain-specific signals [1].

2.1 Language Model (LM) Based Approaches

Text embedding proximity has become a popular approach for company similarity quantification. The key idea is to represent textual descriptions of companies as dense vectors (a.k.a., embeddings) in a high-dimensional space, such that the similarity between two companies can be measured by the proximity between their corresponding vectors. The text embeddings are usually obtained using pretrained LMs, such as BERT [11], T5 [35], LLaMA [43] and GPT-3/4 [5, 32]. The pretrained LMs can be used directly, or finetuned on company descriptions in a supervised [10, 37], semi-supervised [6], or Self-Supervised Learning (SSL) paradigm [15] to further improve the performance of proximity search. Supervised and semi-supervised methods typically yield superior performance (than SSL) on domain-specific tasks, when high-quality annotations are available[6]. This limits its applicability in scenarios where the annotation is scarce, noisy, or costly to obtain.

***N*-shot prompting**. The recent rapid development of LLMs such as GPT-3/4 [5, 32], LLaMA [43], LaMDA [42] and Gemini [36] has led to significant improvements in general-purpose conversational AI like ChatGPT [33]. By prompting them with $N \geq 0$ examples, the instruction tuned models can answer questions about identifying similar companies, for example "*Can you name 10 companies that are similar to EQT?*" As a result, *N*-shot prompting has emerged as a potential tool for investment professionals (e.g., [49, 50]) looking to conduct similar company search and analysis. However, this approach is currently limited by several factors: (1) to ensure the model's responses are up-to-date and relevant, a large amount of domain-specific information must be incorporated via RAG (retrieval augmented generation) alike solutions; (2) the decision-making process may not be explainable, making it difficult to understand and trust the answer; (3) the performance depends heavily on prompt design.

2.2 Knowledge Graph (KG) Based Approaches

Companies can be represented as nodes in a graph, where each node is enriched with attributes. There can be various types of similarities between any two companies, which can be naturally represented as heterogeneous edges in the graph. Moreover, the strength of each relation can be captured by assigning appropriate edge weights. Such a graph-based representation enables the use of heuristic algorithms or GNN models for searching and identifying similar companies, based on their structural and attribute-level similarities. According to our survey[8], most public graph datasets are designed for predicting node [52], edge [3] or graph [39] level properties, while our graph is tailored for company similarity quantification[9]. The most common entities in KGs are web pages [31], papers [2], particles [4], persons [26] and so on. Relato Business Graph[10], which also represents companies as node entities, is the closest dataset to ours. However, its limited scale and absence of edges implying company similarity could constrain its suitability for quantifying company similarity. To address this, we build CompanyKG, a large-scale company KG that incorporates diverse and weighted similarity relationships in its edges. We present extensive benchmarking results (over four carefully assembled evaluation tasks) for 11 reproducible baselines categorized into three groups: node-only, edge-only, and node+edge.

2.3 KG + LM Approaches

Recently, there has been a growing effort to integrate the strengths of LMs and KGs. This integration can be broadly categorized into three main approaches [12, 34]:

(1) **Self-supervised training of GNN and LM for relation prediction**: Methods such as SimKGC [47] and others [27] focus on training GNNs and LMs together in a self-supervised manner to enhance relation prediction within the graph.
(2) **Graph RAG (Retrieval Augmented Generation)**: This approach involves augmenting the prompting process of pretrained LMs with relevant sub-graphs retrieved from the KG. Examples include techniques discussed in [29] and [14]. These methods typically convert the sub-graph into a text format that can be processed by the LM, enhancing its reasoning capabilities with structured graph information.
(3) **Joint finetuning of pretrained LM and GNN**: A pretrained LLM and a pretrained GNN are jointly finetuned to create a multimodal LLM capable of performing complex question-answering (QA) tasks. Examples include GreaseLM [54] and GraphTranslator [53]. These models use a mechanism to retrieve relevant sub-graphs and encode them into the same language space as the LLM, allowing for integrated reasoning over both text and graph modalities.

This field is highly active in EQT Motherbrain, where we are developing an investor-centric QA agent leveraging a fine-grained spatio-temporal KG[11]. This KG encompasses a diverse set of entity types, including companies, persons, industry sectors, products, deals, and more. Given that CompanyKG is specifically designed for company similarity quantification, the remainder of this paper will primarily focus on LM and KG-based approaches.

[7]Diffbot: www.diffbot.com; Pitchbook: www.pitchbook.com; Crunchbase: www.crunchbase.com; CB Insights: www.cbinsights.com; S&P Capital IQ: www.capitaliq.com;. These data sources can be used beyond similar company quantification.

[8]We have reviewed graph datasets in SNAP [25], OGB [22], Network Repository [38], Hugging Face (https://huggingface.co), Kaggle (https://www.kaggle.com) and Data World (https://data.world)

[9]Formally, it is equivalent to edge prediction, yet edge prediction methods like GraphSAGE usually assume an exhaustive edge representation in the graph, which is not the case in CompanyKG as discussed in Section 3.1.

[10]https://data.world/datasyndrome/relato-business-graph-database

[11]Private Equity KG (PEKG) will be introduced here: https://motherbrain.ai

3 CompanyKG (V2)

3.1 Edges (Relations): Types and Weights

We model 15 different inter-company relations as undirected edges, each of which corresponds to a unique edge type (ET) numbered from 1 to 15 as shown in Table 1. These ETs capture six widely adopted categories (C1∼C6) of similarity between connected company pairs: C1 - competitor landscape (ET2, 5 and 6), C2 - industry sector (ET3, 4, 9 and 10), C3 - M&A transaction (ET1, 7 and 11), C4 - people's affiliation (ET8 and 15), C5 - news/events engagement (ET12 and 14), and C6 - product positioning (ET13). Among them, ET1, 2, 3, 7 and 10 are aligned with EQT's scope to varying degrees, while the other 10 ETs are not bound to EQT's scope at all, as they are sourced from well-known external data providers such as Pitchbook, Crunchbase, and LinkedIn. The constructed edges do not represent an exhaustive list of all possible edges due to incomplete information (Appx. D-B-3 [8]). Consequently, this leads to a sparse and occasionally skewed distribution of edges for individual ETs, posing additional challenges for downstream learning tasks.

Associated with each edge of a certain type, we calculate a real-numbered weight as an approximation of the similarity level of that type. The edge definition, distribution (both absolute quantity and percentage), and weights calculation are elaborated in Table 1. To provide insight into the distribution of edge weights, we present histograms of the weights for all 15 ETs (ET1-15) in Figure 2: while some edge types (ET1-6 and ET11) have discrete weights, others follow a continuous distribution. As depicted in Figure 1, we merge edges between identical company pairs into one, assigning a 15-dim weight vector. Here, the i-th dimension signifies the weight of the i-th ET. Importantly, a "0" in the edge weights vector indicates an "*unknown relation*" rather than "*no relation*", which also applies to the cases when there is no edge at all between two nodes. Although there are a few ETs that can naturally be treated as directed (e.g. ET7, M&As), most are inherently undirected. We therefore treat the whole graph as undirected.

3.2 Nodes (Companies): Features and Profiles

The CompanyKG graph includes all companies connected by edges defined in the previous section, resulting in a total of 1,169,931 nodes[12]. Each node represents a company and is associated with a descriptive text, such as "*Klarna is a fintech company that provides support for direct and post-purchase payments ...*". As mentioned in Appx. D-B [8], about 5% texts are in languages other than English. To comply with privacy and confidentiality requirements, the true company identities or any raw interpretable features of them can not be disclosed. Consequently, as demonstrated in the top-right part of Figure 1, we encode the text into numerical embeddings using four different pretrained text embedding models: mSBERT (multilingual Sentence BERT) [37], ADA2[13], SimCSE [15] (fine-tuned on textual company descriptions) and PAUSE [6] (specifically, the "PAUSE-SC-10%" model in Section 4.4 of [6]). The choice of embedding models encompasses both pre-trained, off-the-shelf (mSBERT and ADA2) and proprietary fine-tuned (SimCSE and PAUSE) models. To showcase the sophistication and inclusiveness of the company profiles, we collect five attributes (a snapshot as of February 2023) for each company. The attributes have different rate of missing values and are bucketed. The attributes include (1) *employee* (Figure 3a): the total number of employees; (2) *funding* (Figure 3b): the accumulative funding (in USD) received by the company; (3) *geo_region* (Figure 3c): the geographic region where the company is registered; (4) *duration* (Figure 3d): the number of years since the company was founded; (5) *sector* (Figure 3e): the top level of industry sectors [7] of the company. As illustrated in Figure 3, the companies are distributed widely over each attribute, demonstrating the graph's sophistication and inclusiveness.

[12]Due to the company relations originating from various data sources, leading to duplicated company entities, a company entity resolution system has been implemented to ensure node uniqueness. For more details, see Appx. A.1

[13]ADA2 is short for "text-embedding-ada-002", which is OpenAI's most recent embedding engine employing a 1536-dimensional semantic space, with default settings used throughout the experiments.

3.3 Evaluation Tasks

The main objective of CompanyKG is to facilitate developing algorithms/models for recommending companies similar to given seed/query companies. The effectiveness of such recommendations relies on the capability of quantifying the degree-of-similarity or predicting the relation between any pair of companies. Two-stage paradigms are prevalent in large-scale recommender systems [46]: the first stage swiftly generates candidates that fall into the similarity ballpark, while the second stage ranks them to produce the final recommendations. We assemble a Competitor Retrieval (CR) task from real investment deals to assess the end-to-end performance over the entire task of company expansion. To understand the performance for each stage, we carefully curate two additional evaluation tasks: Similarity Prediction (SP) and Similarity Ranking (SR). In fact, effectively extrapolating CompanyKG's relations/edges into the future is crucial for performing well on the CR, SP, and SR tasks. To this end, we introduce an additional Edge Prediction (EP) evaluation task to provide researchers with deeper insights into the performance and limitations of their models.

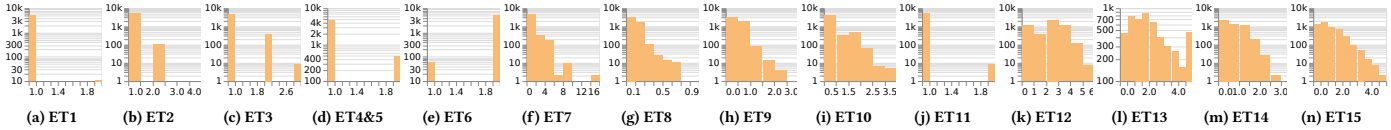
Competitor Retrieval (CR) describes how a team of experienced investment professionals would carry out competitor analysis, which will be a key application of a company expansion system. Concretely, investment professionals perform deep-dive investigation and analysis of a target company once they have identified it as a potential investment opportunity. A critical aspect of this process is competitor analysis, which involves identifying several direct competitors that share significant similarities (from perspectives of main business, marketing strategy, company size, etc.) with the target company. EQT maintains a proprietary archive of deep-dive documents that contain competitor analysis for many target companies. We select 76 such documents from the past 4 years pertaining to distinct target companies and ask human annotators to extract direct competitors from each document, resulting in ∼5.3 competitors per target. Ideally, for any target company, we expect a competent algorithm/model to retrieve all its annotated direct competitors and prioritize them at the top of the returned list.

Similarity Prediction (SP) defines the coarse binary similar-vs.-dissimilar relation between companies. Excelling in this task often translates to enhanced recall in the first stage of a recommender system. It also tends to be valuable for use cases like market mapping, where an inclusive and complete result is sought after. For SP task, we construct an evaluation set comprising 3,219 pairs of companies that are labeled either as positive (similar, denoted by

Type	The specification of each edge type (ET) and the associated weight	#edge	%
ET1 (C3)	In Motherbrain platform, investment professionals (i.e., users) can create a so-called "space" by specifying a theme such as "*online music streaming service*". In each space, users are exposed to some companies that are recommended (by a linear model) as theme-compliant. Users can approve or reject each recommended company, which will be used to continuously train the model in an active learning fashion. We create an edge between any two confirmed companies in the same space. For any two companies with an edge with this type, the associated edge weight counts their approval occurrences in all spaces.	117,924	0.23
ET2 (C1)	Motherbrain users can add $N \geq 1$ direct competitors for any target company, resulting in a competitor set containing $N+1$ companies (N direct competitors and the target company itself). For each target company with such a competitor set, we create one fully connected sub-graph. When merging these sub-graphs, the duplicated edges are merged into one with a weight indicating the degree of duplication.	16,734	0.03
ET3 (C2)	Investment professionals can create "collections" of companies for various purposes such as market mapping and portfolio tracking. We manually selected 787 (out of 1,458 up till February 2023) collections that contain companies with a certain level of similarity. We create a fully connected sub-graph for each collection. These sub-graphs are then merged and deduplicated like **ET2**, so the edge weights represent the duplication degree.	374,519	0.73
ET4 (C2)	Based on a third-party data source* containing about 4 million company profiles, we create graph edges from company market connections. The edge weights represent the count of the corresponding market connection in that data source.	1,473,745	2.89
ET5 (C1)	From a third-party data source* about companies, investors and funding rounds, we extract pairwise competitor relations and build edges out of them. Again, the edge weights are the number of occurrences of such extracted relation.	74,142	0.15
ET6 (C1)	Same as **ET5** except that competitor information is extracted from a different data source* that specializes in financial data from private and public companies.	346,162	0.68
ET7 (C3)	Motherbrain, as a team, also work with deal teams on various M&A and add-on acquisition† projects. Such projects aim to produce a list of candidate companies for a certain company to potentially acquire. For each selected project* up till December 2022, we create a fully connected sub-graph from a company set containing the acquirer and all acquiree candidates. These sub-graphs are then merged and deduplicated like **ET2** and **ET3**, so the edge weights represent the duplication degree.	55,366	0.11
ET8 (C4)	Based on the naïve assumption that employees tend to move among companies that have some common traits, we build graph edges representing how many employees have flowed (since 1990) between any two companies. For any two companies A and B who currently have N_A and N_B employees respectively, let $N_{A:B}$ and $N_{B:A}$ respectively denote the number of employees flowed from A to B and from B to A. We create an edge between A and B either when (1) $N_{A:B}+N_{B:A} \geq 20$, or (2) $N_{A:B}+N_{B:A} \geq 3$ and $(N_{A:B}+N_{B:A})/(N_A+N_B) \geq 0.15$. We assign the value of $(N_{A:B}+N_{B:A})/(N_A+N_B)$ as the edge weight.	71,934	0.14
ET9 (C2)	In Motherbrain's data warehouse, each company has some keywords (e.g., company EQT has three keywords: *private equity*, *investment* and *TMT*) that are integrated from multiple data sources*. We filter out the overly short/long keywords and obtain 763,503 unique keywords. Then we calculate IDF (inverse document frequency) score for every keyword and apply min-max normalization. Finally, we create an edge between any two companies that have shared keyword(s), and assign the average (over all shared keywords) IDF score as the corresponding edge weight.	45,717,108	89.57
ET10 (C2)	EQT has defined a hierarchical sector framework [7] to support thematic investment activities. Motherbrain users can tag a company as belonging to one or more sectors (e.g., *cyber security*, *fintech*, etc.). For each low-level sector (i.e., sub-sector and sub-sub-sector), we create a fully connected sub-graph from all the tagged companies. The sub-graphs are further merged into one big graph with edge weights indicating duplication degree of edges. It also worth mentioning that an edge created from a sub-sub-sector is weighted twice as important as the one created from a sub-sector.	813,585	1.59
ET11 (C3)	Mergers and/or acquisitions could imply a certain level of similarity between the acquirer and acquiree. As a result, we create an edge between the involved companies of historical (since 1980) merger/acquisition (specifically M&A, buyout/LBO, merger-of-equals, acquihire)‡ events. The edge weight is the number of occurrences of the company pair in those events.	260,644	0.51
ET12 (C5)	Based on a third-party data source* that keeps track of what events (e.g., conferences) companies have attended, we create a preliminary edge between two companies who have had at least five co-attendance events in the past (up till June 2022). Then, we further filter the edges by computing the Levenshtein distance between the company specialties that come with the data source in textual strings. The edge weights are the count of co-attendance in log scale.	1,079,304	2.11
ET13 (C6)	From a product comparison service*, we obtain data about what products are chosen (by potential customers) to be compared. By mapping the compared products to the owning companies, we can infer which companies have been compared in the past up till February 2023. We build a graph edge between any two companies whose products has been compared. The edge weights are simply the number of times (in log scale) they are compared.	216,291	0.42
ET14 (C5)	Motherbrain has a news feed* about global capital market. Intuitively, companies mentioned in the same piece of news might be very likely connected, thus we create an edge between any pair of the co-mentioned companies (till February 2023). When one of the co-mentioned companies is an investor, the relation is often funding rather than similarity, therefore we eliminate those edges from the final graph. The edge weights are log-scale co-mention count.	151,302	0.30
ET15 (C4)	One individual may hold executive roles in multiple companies, and it is possible for them to transition from one company to another. When any two companies, each having fewer than 1,000 employees, share/have shared the same person in their executive role, we establish a graph edge between them. To address the weak and noisy nature of this similarity signal, we refine the edges by only retaining company pairs that share at least one keyword (cf. **ET9**). The edge weights are log-scale count of shared executives between the associated companies.	273,851	0.54

* The detailed information is hidden due to legal and compliance requirements. Our intent is not to re-license any third-party data; instead, we focus on sharing aggregated and anonymized data for pure research purposes only. Please refer to Appx. D-E in [8] for more information about dataset usage.
† An add-on acquisition is a type of acquisition strategy where a company acquires another company to complement and enhance its existing operations or products.
‡ Leveraged buyout (LBO) is a transaction where a company is acquired with a significant loan. Acquihire is the process of acquiring a company primarily to recruit its employees, rather than to gain control of its products/services. A merger-of-equals is when two companies of about the same size come together to form a new company.

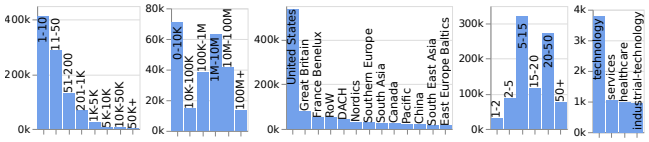
Table 1: Specification of 15 edge types (ET1~ET15), edge weights (stronger relations has higher weights), and the number of edges per type ("#edge" column). ETs fall into six extensively recognized categories of company similarity: C1 (competitor landscape), C2 (industry sector), C3 (M&A transaction), C4 (people's affiliation), C5 (news/events engagement), and C6 (product positioning).



(a) ET1 (b) ET2 (c) ET3 (d) ET4&5 (e) ET6 (f) ET7 (g) ET8 (h) ET9 (i) ET10 (j) ET11 (k) ET12 (l) ET13 (m) ET14 (n) ET15

Figure 2: The weight distributions of different ETs: we randomly sample 5,000 edges for each type. The histograms of ET4 and ET5 are almost identical, hence they are merged.

"1") or negative (dissimilar, denoted by "0"). Of these pairs, 1,522 are positive and 1,697 are negative. Positive company pairs are inferred from user interactions with our Motherbrain platform – closely related to ET2&3, although the pairs used in this test set are selected from interactions after the snapshot date, so we ensure there is no leakage. Each negative pair is formed by randomly sampling one



(a) employee (b) funding (c) geo_region. (d) duration (e) sector

Figure 3: The distribution (y-axis) of bucketed company/node attributes (x-axis). In (c), "France Benelux" refers to France, Netherlands, Belgium and Luxembourg; "DACH" includes Germany, Austria and Switzerland; "RoW" stands for Rest of the World.

company from a Motherbrain collection (cf. ET3 in Table 1) and another company from a different collection. All samples have been manually examined by domain experts.

Similarity Ranking (SR) is designed to assess the ability of any method to rank candidate companies (numbered 0 and 1) based on their similarity to a query company. Excelling in this task is of utmost significance for the second stage of a high-quality recommender system. When applied to real-world investment scenarios like M&A, a fine-grained ranking capability can significantly enhance the efficacy of prioritizing M&A candidates. During the creation of this evaluation dataset, we need to ensure a balanced distribution of industry sectors and increase the difficulty level of the task. Therefore, we select the query and candidate companies using a set of heuristics based on their sector information [7]. As a result, both candidates will generally be quite closely related to the target, making this a challenging task. As described in Appx. A.2, paid human annotators, with backgrounds in engineering, science, and investment, were tasked with determining which candidate company is more similar to the query company. Each question was assessed by at least three different annotators, with an additional annotator involved in cases of disagreement. The final ground-truth label for each question was determined by majority voting. This process resulted in an evaluation set with 1,856 rigorously labeled ranking questions. We retained 20% (368 samples) as a validation SR set for model selection. The choice to use the SR task for model selection, instead of the SP or CR tasks, is primarily because it covers a more representative sample of the entire graph. See Appx. D-B-8 [8] for further explanation.

Edge Prediction (EP) is designed to evaluate how effectively a model can predict missing or future relationships between any two companies. Successfully achieving this is akin to extending the CompanyKG into the future, allowing investment professionals to gain fine-grained, forward-looking insights in all aforementioned CR, SP, and SR tasks. Specifically, we collected relations[14] as described in Table 1 that were first ingested into our data warehouse between April 6, 2023, and May 25, 2024. Consequently, the edges in the EP dataset are not present in CompanyKG, which is a snapshot up until April 5, 2023. To simplify the evaluation procedure, we (1) selected only the edges whose nodes are both present in CompanyKG, and (2) binarized the edge weights (exist vs. missing) to form a multi-class, single-label classification problem. To balance the samples in each edge type (ET), we randomly sampled 5,000 edges for inclusion in the EP dataset, resulting in a total of 40,000 samples. For each ET, we reserved 1,500 samples for validation and used the remaining samples to report test performance.

[14]Due to changes in data source availability after April 5, 2023, we collected new edges for ET2, ET3, ET4, ET5, ET8, ET9, ET14, and ET15.

4 Experiments

In addition to the CompanyKG dataset, we provide comprehensive benchmarks from 11 popular and/or state-of-the-art baselines categorized into three groups: node-only, edge-only, and node+edge.

Node-only baselines use only the features of individual companies provided in the graph and ignore the graph structure. We measure the cosine similarity between the embeddings of a given type (see Section 3.2) that are associated with any two companies in the graph. This gives us an embedding proximity score, which can be evaluated in the SP task, and a method of ranking candidates, evaluated by SR and CR. In order to assess the cutting-edge N-shot prompting methodology, we transform each raw SP/SR/CR evaluation sample into a text prompt and let ChatGPT [33] generate the label. The prompt templates are provided in Appx. B-E [8].

Edge-only baselines merely utilize the graph structure. We define four simple heuristics to rank companies C_i by proximity to a target company T: (1) *Unweighted Shortest Path* (*USP*) length from T to C_i; (2) total weight of the *Weighted Shortest Path* (*WSP*) from T to C_i (weights inverted so lower is more similar); (3) close proximity only if T and C_i are immediate *Neighbors*; and (4) rank weighted neighbors (abbreviated as *W. Neighbors*) – immediate neighbors to T ranked by edge weight. All heuristics will choose randomly if not discriminative. To allow comparison of edge and path weights from different ETs, we scale edge weights to $[0, 1]$ for ET and shift them so they have a mean weight of 1. In order to avoid edge weights playing too strong a role in WSP compared to presence/absence of edges and paths, we add a constant[15] to all weights.

Node+edge baselines leverage both the node feature and graph structure. CompanyKG does not provide an explicit signal that can guide supervised learning to combine these into a single company representation for similarity measurement. As a result, we test five popular self-supervised GNN methods: **GRACE** [56], **MVGRL** [18],**GraphSAGE** [17], **GraphMAE** [21] and **eGraphMAE** [8]. GRACE and MVGRL are graph contrastive learning (GCL) approaches that construct multiple graph views via stochastic augmentations before learning representations by contrasting positive samples with negative ones [55]. GraphSAGE, which is especially suited for edge prediction, is trained to ensure that adjacent nodes have similar representations, while enforcing disparate nodes to be distinct. In light of the recent success of generative SSL in Natural Language Processing (NLP) and Computer Vision (CV) domains, Hou et al. [21] proposed GraphMAE, a generative SSL method for graphs that emphasizes feature reconstruction. To incorporate the edge weights in GraphMAE, Cao et al. [8] replaced the graph attention network (GAT) [44] with an *edge-featured* GAT (eGAT) [48], resulting in eGraphMAE (edge-featured GraphMAE). Unless otherwise specified, we report the best results from a hyper-parameter grid search detailed in Appx. B, using the validation split of the SR task set described in Section 3.3.

[15]We set this constant to 5 after a few empirical explorations. Results could potentially be improved by tuning this constant, but the method is intended here only as a simple baseline.

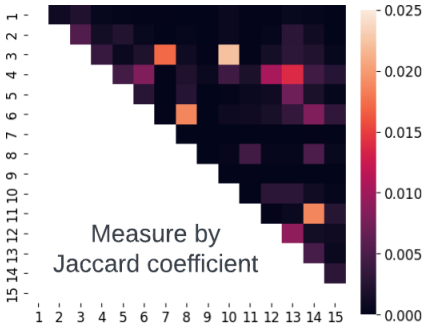


Figure 4: Pairwise redundancy analysis of ETs.

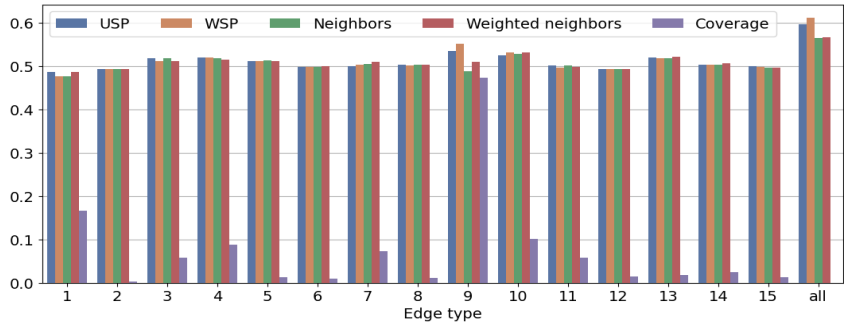


Figure 5: Ranking performance (SR test set) of different ETs using USP and WSP heuristics. Coverage for "all" is 100%.

4.1 Comparison of ETs (Edge Types)

Before presenting the results of the evaluations on the full graph, we intend to gain some insights into the differences between the 15 different ETs. To understand the extent of redundancy among different ETs, we measure the overlap between any two ETs, such as ETi and ETj. Letting $\{ET i\}$ and $\{ET j\}$ be the set of unweighted edges belonging to ETs i and j respectively, we measure the Jaccard coefficient $J = |\{ET i\} \cap \{ET j\}| / |\{ET i\} \cup \{ET j\}|$, where $|\cdot|$ denotes the size of the embodied set. The heatmap in Figure 4 shows that for most pairs of ETs, there is effectively no overlap, whilst for a few (e.g., 3–10, 11–14, 6–8) there is a slightly stronger overlap. However, the coefficient for the most overlapped pair, 3–10, is still very small ($J = 0.022$), suggesting that the information provided by the different ETs is complementary.

To investigate further, we create a sub-graph for each individual ET by preserving only edges of that type, and apply the four edge-only heuristics to evaluate on the SR task (test set) using only that sub-graph. SR is chosen because of its superior representativeness of companies and alignment with our model selection approach (Section 4 and Appx. B). From Figure 5, we see some variation between the performance of different heuristics using different ETs. On many ETs, performance of all heuristics is close to random, since coverage is low (1%-47%), and few are able to yield performance much above chance. For ET9 (highest coverage), the full-path methods, USP and WSP, perform considerably better than immediate neighbors, but the same pattern is not seen, for example, on ET10 (also relatively high coverage). Nevertheless, the best performance is seen when all ETs are used together, giving full coverage of the companies in the samples, suggesting again the complementary nature of ETs. Here we see that full-path heuristics perform much better than immediate neighbors, reflecting useful structural information in the graph beyond the company-company relations that constitute the edges of the graph. Making use of the weights in WSP helps further, suggesting that they also carry useful information.

4.2 Performance of SP (Similarity Prediction)

We apply the immediate neighbor graph heuristics by classifying all pairs of companies that are directly connected by an edge in the graph as similar and others as dissimilar. We extend this to the weighted case by scaling weights to a 0-1 range and using them as similarity scores where edges exist. Results are reported in Table 2. Edge-only baselines perform poorly: although the test data is similar in nature certain ETs in the graph, there is no overlap between the test pairs and the graph edges, so such a simple heuristic is not sufficient. Node-only baselines span a broad range, the best achieving ~0.8, which is comparable to ChatGPT and GNN-based methods. Seen from the full results in Table 4, the de facto highest SP performance (0.8253) is achieved by eGraphMAE+SimCSE, showing a slight improvement brought by incorporating edge weights.

4.3 Performance of SR (Similarity Ranking)

To evaluate SR task for KG and embedding proximity methods, we obtain embeddings for the query company and two candidate companies using each method. Next, we calculate cosine similarity scores between the query company and the two candidate companies resulting in two scores. The final prediction of the more similar company is made by choosing the candidate with the higher score. Evaluation using edge-only graph heuristics is described in Section 4.1. We report the SR accuracy in Table 2. Edge-only methods establish a baseline of ~60%, while some node-only methods (e.g., mSBERT and ADA2) are able to improve on this (~65%), perhaps due to pretraining on extensive multilingual corpora. These baseline results suggest that a combination of node and edge information into a single representation might be able to take advantage of both sources of similarity prediction. However, the GNN-based methods struggle to improve on the baselines. The best node+edge method, GraphMAE, manages to slightly outperform the best node-only baseline, achieving the overall best result. Incorporating edge weights into the model (eGraphMAE) does not succeed in improving on this result, even though we found using graph heuristics that the edge weights carry useful information.

4.4 Performance of CR (Competitor Retrieval)

For the CR task, we aim to determine how likely the direct competitors of a target company are to appear in the top-K most similar companies. To accomplish this, we perform an exhaustive search in the embedding space and return the top-K companies with the highest cosine similarity scores to the target company. We then count the number of annotated direct competitors (denoted as M) that appear within the returned set and compute the metric of Recall@K, which is given by $\frac{M}{N}$, where N is the total number of direct competitors in the CR test set. Although we test eight K values over a wide range – 50, 100, 200, 500, 1,000, 2,000, 5,000 and 10,000 (cf. Table 4 in Appx. and Figure 6) – it is the lower end of this range that is of most interest for downstream applications: in

Approach / Model		SP (AUC)	SR (Acc%)	CR Recall@50	CR Recall@100
Graph heuristics	USP (Unweighted SP)	N/A*	59.61	18.27	29.48
	WSP (Weighted SP)	N/A*	61.16	**43.69**	**56.03**
	Neighbors	0.6229†	56.52	22.25	31.84
	W. Neighbors	0.6020	56.65	43.50	54.65
Embedding proximity	mSBERT (512-dim)	**0.8060**	67.14	12.96	18.24
	ADA2 (1536-dim)	0.7450	67.20	14.09	21.69
	SimCSE (768-dim)	0.7188	61.69	7.66	8.90
	PAUSE (32-dim)	0.7542	65.19	6.84	9.62
N-shot	ChatGPT-3.5	0.7501†	66.73†	30.06	31.10
Knowledge Graph (GNN-based)	GraphSAGE (ADA2)	0.7422 ±0.0202	62.90 ±2.25	10.12 ±3.91	11.93 ±0.96
	GRACE (mSBERT)	0.7243 ±0.0233	59.36 ±0.64	2.68 ±1.82	4.64 ±1.51
	MVGRL (PAUSE)	0.6843 ±0.0280	55.60 ±1.18	0.47 ±0.35	1.25 ±0.56
	GraphMAE (mSBERT)	0.7981 ±0.0063	**67.61** ±0.11	20.88 ±0.46	27.83 ±0.39
	eGraphMAE (mSBERT)	0.7963 ±0.0030	67.52 ±0.03	18.44 ±0.21	23.79 ±0.22

* We omit SP evaluation for the path-based graph heuristics, since, whilst they are able to rank companies by similarity, there is no obvious way to obtain a 0-1 similarity score for a pair of companies from them.
† To account for the binary nature of the SP prompts answered by ChatGPT and the Neighbors heuristic, we report accuracy instead of AUC.

Table 2: The performance of baselines on three evaluation tasks (SP, CR and SR) of CompanyKG. Best results are in bold. Standard deviations are reported over 3 random initializations of the trained models.

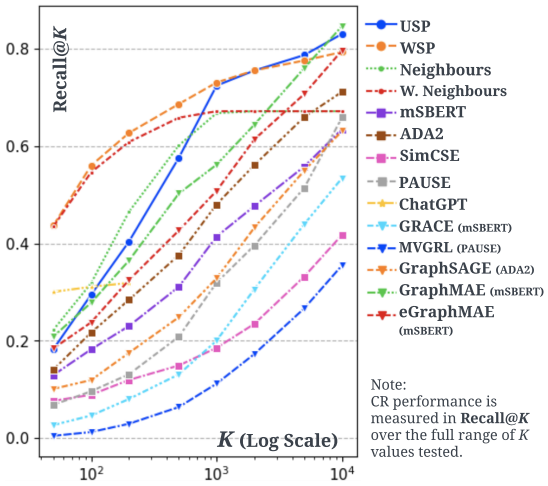


Figure 6: Performance comparison on CR task measured in Recall@K over the full range of K values tested.

any search/retrieval system, the user will only in practice be able to examine a small number of the top-ranked results. We therefore only report Recall@50 and @100 in Table 2.

Edge-only heuristic methods, specifically WSP and W. Neighbors, clearly stand out as superior. This underscores the significance of both the graph structure and edge weights, as we saw with graph heuristics in Section 4.1. The generative graph learning methods, GraphMAE and eGraphMAE, yield robust results, comparable to USP and Neighbors, suggesting that they succeed in capturing some salient aspects of the graph structure, but not in exploiting the edge weights. However, other GNN-based methods like GRACE perform poorly across all Ks in comparison to edge-only and most node-only baselines. Whilst the comparative performance of the methods is similar to the other tasks, the differences are greater in CR. This suggests CR is a challenging task and may benefit greatly from KG-based learning, making it suited for benchmarking a model's ability to harness node and edge simultaneously.

4.5 Performance of EP (Edge Prediction)

In evaluating EP, we assume the availability of numerical node/company embeddings (either learned or provided) as input features for predicting the existence of edges in the EP dataset. Our default setup involves randomly sampling a balanced training dataset (from CompanyKG), formatted identically to the EP evaluation dataset. For each edge in the training/validation/test dataset, we concatenate the embeddings of the two end nodes to form an input feature vector. We train a 3-layer neural network with the architecture of Linear(512)-RELU-Dropout(0.5)-Linear(256)-RELU-Dropout(0.5)-Linear(8), which performs 8-class classification (one class for each included ET). Training is conducted for up to 100 epochs, utilizing an early stopping mechanism leveraging the EP validation split. If the performance (AUC over all ETs) on the validation split does not improve for more than 5 consecutive epochs, training is stopped. The model with the best validation performance is then used for evaluation on the test split.

In Table 3, we present the overall AUC and ET-specific AUCs on the EP test split, averaged over three trials with their respective standard deviations. We evaluate four LM embeddings (derived from company descriptions) and five GNN embeddings for each LM embedding, resulting in a total of 20 different GNN embeddings. Regarding hyperparameters, we use the optimized settings detailed in Appx. B. We can see that GraphMAE and eGraphMAE exhibit the best performance. Notably, eGraphMAE excels in the majority of ETs, likely due to its ability to learn from edge weights.

4.6 Embedding Space Visualization

Intuitively, it is desirable for similar companies to cluster closely together in the embedding space, while dissimilar ones are preferably pulled apart. To gain qualitative insights from this perspective, we reduce the embeddings (from various baseline methods) to two dimensions using UMAP [30], and color-code them by manually annotated sectors (cf. Figure 3e). We visualize the embeddings before (e.g., mSBERT) and after GNN training in Figure 7. It is noticeable that GNN-based methods tend to nudge companies within the same sector closer together, which could help with tasks like industry sector prediction [7]. However, this improved embedding clustering has no clear positive correlation to the performance of the evaluation tasks designed for similar company quantification.

5 Conclusion and Discussion

To represent and learn diverse company features and relations, we propose a large-scale heterogeneous graph dataset—CompanyKG (V2), originating from a real-world investment platform. Specifically, 1.17 million companies are represented as nodes enriched with company description embeddings, and 15 different inter-company relations result in 51.06 million weighted edges. Additionally, we carefully compiled four evaluation tasks (SP, CR, SR, and EP) and presented extensive benchmarking results for 11 reproducible baselines categorized into three groups: node-only, edge-only, and node+edge. While node-only methods show good performance in the SR task,

Embedding Model			Overall AUC	ET2 AUC	ET3 AUC	ET4 AUC	ET5 AUC	ET8 AUC	ET9 AUC	ET14 AUC	ET15 AUC
Embedding from LMs		mSBERT	0.8172 ±0.0024	0.7544 ±0.0033	0.8138 ±0.0079	0.7566 ±0.0047	0.9265 ±0.0009	0.9263 ±0.0003	0.7627 ±0.0084	0.7885 ±0.0030	0.8086 ±0.0022
		ADA2	0.8164 ±0.0055	0.7488 ±0.0225	0.8023 ±0.0071	0.7660 ±0.0014	0.9141 ±0.0026	0.9400 ±0.0004	0.7595 ±0.0059	0.8021 ±0.0029	0.7980 ±0.0112
		SimCSE	0.8014 ±0.0021	0.7475 ±0.0130	0.7910 ±0.0044	0.7227 ±0.0037	0.9159 ±0.0011	0.9178 ±0.0014	0.7346 ±0.0002	0.7653 ±0.0035	0.8163 ±0.0023
		PAUSE	0.7692 ±0.0015	0.7972 ±0.0171	0.5944 ±0.0078	0.7771 ±0.0040	0.9024 ±0.0028	0.8361 ±0.0059	0.6986 ±0.0076	0.7616 ±0.0020	0.7860 ±0.0029
Knowledge Graph: GNN-based Embedding (initiated with different LM node embeddings)	mSBERT	GraphSAGE	0.8092 ±0.0025	0.7831 ±0.0096	0.7676 ±0.0093	0.7939 ±0.0025	0.9077 ±0.0031	0.9096 ±0.0017	0.7201 ±0.0196	0.7641 ±0.0038	0.8274 ±0.0016
		GRACE	0.7634 ±0.0089	0.7565 ±0.0242	0.6811 ±0.0094	0.7861 ±0.0082	0.8813 ±0.0073	0.8875 ±0.0018	0.6453 ±0.0152	0.7012 ±0.0283	0.7685 ±0.0111
		MVGRL	0.7356 ±0.0022	0.7697 ±0.0115	0.6228 ±0.0098	0.7431 ±0.0034	0.8491 ±0.0040	0.8476 ±0.0041	0.6217 ±0.0062	0.6781 ±0.0130	0.7526 ±0.0045
		GraphMAE	0.8350 ±0.0004	0.7482 ±0.0101	**0.8151** ±0.0059	0.8150 ±0.0048	0.9342 ±0.0011	0.9301 ±0.0012	0.7928 ±0.0152	0.8032 ±0.0025	0.8417 ±0.0003
		eGraphMAE	0.8355 ±0.0007	0.7565 ±0.0113	0.8140 ±0.0065	0.8170 ±0.0050	0.9320 ±0.0014	0.9280 ±0.0016	0.7910 ±0.0145	0.8000 ±0.0030	0.8390 ±0.0005
	ADA2 *	GraphSAGE	0.8110 ±0.0030	0.7815 ±0.0102	0.7650 ±0.0085	0.7955 ±0.0031	0.9060 ±0.0028	0.9105 ±0.0102	0.7220 ±0.0185	0.7620 ±0.0041	0.8285 ±0.0020
		GRACE	0.7760 ±0.0046	0.7289 ±0.0173	0.7137 ±0.0028	0.7907 ±0.0008	0.8851 ±0.0024	0.9011 ±0.0025	0.6922 ±0.0116	0.7175 ±0.0057	0.7790 ±0.0056
		MVGRL	0.7184 ±0.0081	0.7921 ±0.0074	0.5905 ±0.0053	0.7377 ±0.0054	0.8301 ±0.0107	0.8338 ±0.0042	0.5700 ±0.0211	0.6655 ±0.0157	0.7280 ±0.0164
		GraphMAE	0.8434 ±0.0035	0.7542 ±0.0048	0.8117 ±0.0069	0.8229 ±0.0010	0.9437 ±0.0015	0.9444 ±0.0016	0.8090 ±0.0086	**0.8197** ±0.0041	0.8419 ±0.0062
	SimCSE	GraphSAGE	0.8050 ±0.0052	0.7650 ±0.0083	0.7540 ±0.0061	0.7800 ±0.0047	0.8960 ±0.0045	0.9000 ±0.0080	0.7170 ±0.0074	0.7520 ±0.0052	0.8150 ±0.0070
		GRACE	0.7687 ±0.0009	0.7528 ±0.0133	0.6824 ±0.0091	0.7830 ±0.0026	0.8829 ±0.0030	0.8895 ±0.0013	0.6800 ±0.0072	0.7104 ±0.0051	0.7686 ±0.0025
		MVGRL	0.7055 ±0.0025	0.6800 ±0.0082	0.5180 ±0.0165	0.7468 ±0.0035	0.8440 ±0.0016	0.8168 ±0.0104	0.6203 ±0.0115	0.6881 ±0.0088	0.7300 ±0.0083
		GraphMAE	0.8378 ±0.0010	0.7867 ±0.0106	0.7997 ±0.0053	0.8134 ±0.0076	0.9309 ±0.0049	0.9302 ±0.0028	0.7934 ±0.0052	0.7973 ±0.0043	**0.8511** ±0.0025
		eGraphMAE	**0.8522** ±0.0064	0.7857 ±0.0073	0.8140 ±0.0073	**0.8250** ±0.0015	**0.9466** ±0.0068	**0.9487** ±0.0029	**0.8120** ±0.0093	0.8028 ±0.0044	0.8404 ±0.0064
	PAUSE	GraphSAGE	0.7734 ±0.0016	0.7805 ±0.0131	0.6328 ±0.0049	0.7729 ±0.0027	0.8917 ±0.0011	0.8671 ±0.0031	0.6859 ±0.0087	0.7537 ±0.0036	0.8026 ±0.0008
		GRACE	0.7089 ±0.0027	0.6879 ±0.0080	0.5148 ±0.0054	0.7566 ±0.0012	0.8560 ±0.0070	0.7944 ±0.0047	0.6188 ±0.0011	0.7107 ±0.0157	0.7319 ±0.0121
		MVGRL	0.7087 ±0.0085	0.7077 ±0.0151	0.5152 ±0.0064	0.7481 ±0.0065	0.8474 ±0.0069	0.8008 ±0.0101	0.6093 ±0.0125	0.7158 ±0.0076	0.7255 ±0.0119
		GraphMAE	0.7865 ±0.0033	0.7836 ±0.0169	0.6389 ±0.0155	0.7727 ±0.0069	0.9014 ±0.0046	0.8799 ±0.0012	0.7355 ±0.0073	0.7795 ±0.0018	0.8001 ±0.0050
		eGraphMAE	0.8024 ±0.0049	**0.7997** ±0.0053	0.6405 ±0.0120	0.7743 ±0.0075	0.9038 ±0.0051	0.8811 ±0.0078	0.7350 ±0.0081	0.7816 ±0.0025	0.8120 ±0.0066

* eGraphMAE is omitted for ADA2 node embedding due to OOM (out-of-memory).

Table 3: The benchmark results on EP (edge prediction) task measured by overall AUC and per-ET AUCs. Best results are in bold.

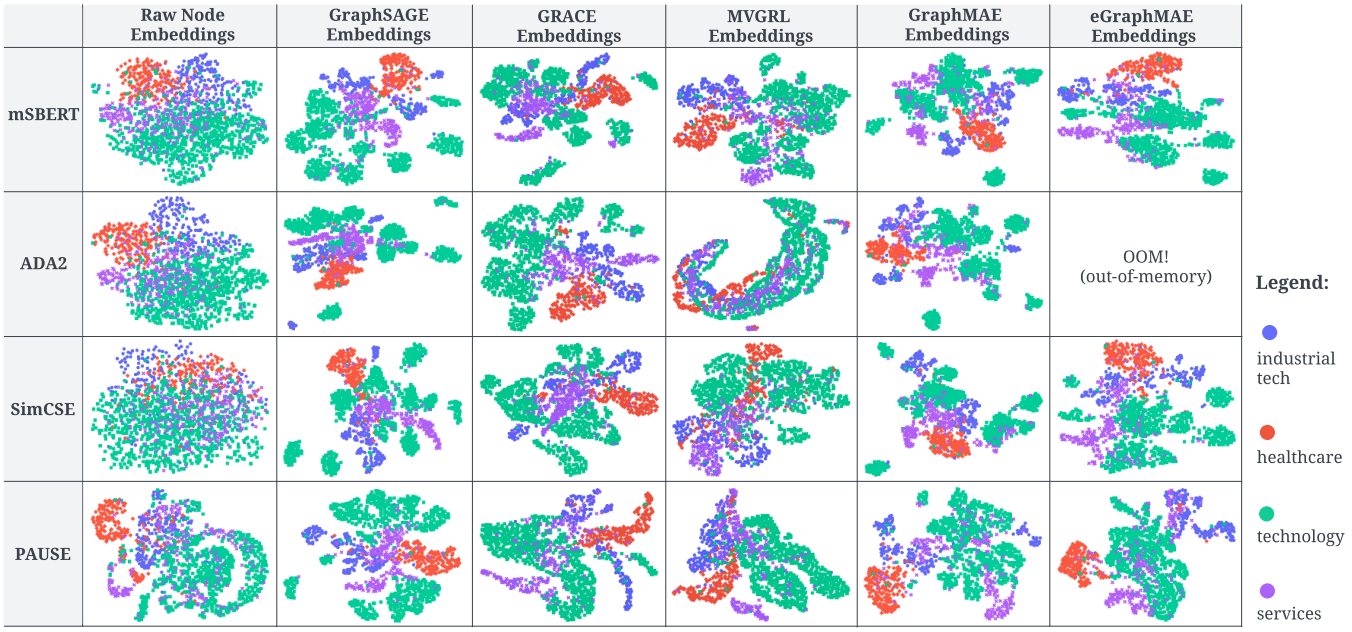


Figure 7: Dimension reduction and visualization of embeddings from different LMs (the first row) and GNN-based models (other rows).

edge-only methods clearly excel in the CR task. For the SP task, node-only and GNN-based approaches outperform edge-only ones by a large margin, with the best result obtained by eGraphMAE, which incorporates node embeddings, edges, and their weights. In the EP task, GraphMAE and eGraphMAE exhibit the best performance, with eGraphMAE excelling in the majority of edge types, likely due to its ability to learn from edge weights. Although generative graph learning exhibits robust performance in general, no single method leads across all CompanyKG tasks, indicating the need for future work to develop more robust learning methods that can effectively incorporate both node and edge information. Overall, CompanyKG can accelerate research on company similarity quantification in the investment domain while serving as a benchmark for assessing self-supervised graph learning methods. Given the continuous evolution of nodes and edges, a logical extension involves capturing KG snapshots over time, thereby transforming it into a spatio-temporal KG.

References

[1] Ryuya Akase, Hiroto Kawabata, Akiomi Nishida, Yuki Tanaka, and Tamaki Kaminaga. 2021. Related Entity Expansion and Ranking Using Knowledge Graph. In *Complex, Intelligent and Software Intensive Systems: Proceedings of the 15th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS-2021)*. Springer, 172–184.

[2] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1ZdKJ-0W

[3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 1247–1250.

[4] Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. 2019. GuacaMol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling* 59, 3 (2019), 1096–1108.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Lele Cao, Emil Larsson, Vilhelm von Ehrenheim, Dhiana Deva Cavalcanti Rocha, Anna Martin, and Sonja Horn. 2021. PAUSE: Positive and Annealed Unlabeled Sentence Embedding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 10096–10107. https://doi.org/10.18653/v1/2021.emnlp-main.791

[7] Lele Cao, Vilhelm von Ehrenheim, Astrid Berghult, Henje Cecilia, Richard Anselmo Stahl, Joar Wandborg, Sebastian Stan, Armin Catovic, Erik Ferm, and Ingelhag Hannes. 2023. A Scalable and Adaptive System to Infer the Industry Sectors of Companies: Prompt + Model Tuning of Generative Language Models. In *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*. 1–11.

[8] Lele Cao, Vilhelm von Ehrenheim, Mark Granroth-Wilding, Richard Anselmo Stahl, Andrew McCornack, Armin Catovic, and Dhiana Deva Cavacanti Rocha. 2023. CompanyKG: A Large-Scale Heterogeneous Graph for Company Similarity Quantification. *arXiv preprint arXiv:2306.10649* (2023).

[9] Lele Cao, Vilhelm von Ehrenheim, Sebastian Krakowski, Xiaoxue Li, and Alexandra Lutz. 2022. Using Deep Learning to Find the Next Unicorn: A Practical Synthesis. *arXiv preprint arXiv:2210.14195* (2022).

[10] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*. 169–174. https://aclanthology.org/D18-2029/

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *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, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[12] Xin Luna Dong. 2023. Generations of Knowledge Graphs: The Crazy Ideas and the Business Impact. *Proceedings of the VLDB Endowment* 16, 12 (2023), 4130–4137.

[13] John Foley, Brendan O'Connor, and James Allan. 2016. Improving Entity Ranking for Keyword Queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. Association for Computing Machinery, New York, NY, USA, 2061–2064.

[14] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1762–1777.

[15] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[17] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[18] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*. PMLR, 4116–4126.

[19] Gerard Hoberg and Gordon Phillips. 2010. Product market synergies and competition in mergers and acquisitions: A text-based analysis. *The Review of Financial Studies* 23, 10 (2010), 3773–3811.

[20] Gerard Hoberg, Gordon Phillips, and Nagpurnanand Prabhala. 2014. Product market threats, payouts, and financial flexibility. *The Journal of Finance* 69, 1 (2014), 293–324.

[21] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. GraphMAE: Self-Supervised Masked Graph Autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 594–604.

[22] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv preprint arXiv:2005.00687* (2020).

[23] Nandish Jayaram, Mahesh Gupta, Arijit Khan, Chengkai Li, Xifeng Yan, and Ramez Elmasri. 2014. GQBE: Querying knowledge graphs by example entity tuples. In *2014 IEEE 30th International Conference on Data Engineering*. 1250–1253. https://doi.org/10.1109/ICDE.2014.6816753

[24] Charles MC Lee, Eric C So, and Charles CY Wang. 2021. Evaluating firm-level expected-return proxies: implications for estimating treatment effects. *The Review of Financial Studies* 34, 4 (2021), 1907–1951.

[25] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[26] Jure Leskovec and Julian Mcauley. 2012. Learning to discover social circles in ego networks. *Advances in neural information processing systems* 25 (2012).

[27] Da Li, Boqing Zhu, Sen Yang, Kele Xu, Ming Yi, Yukai He, and Huaimin Wang. 2023. Multi-task pre-training language model for semantic network completion. *ACM Transactions on Asian and Low-Resource Language Information Processing* 22, 11 (2023), 1–20.

[28] Matteo Lissandrini, Davide Mottin, Themis Palpanas, and Yannis Velegrakis. 2020. Graph-query suggestions for knowledge graph exploration. In *Proceedings of The Web Conference 2020*. 2549–2555.

[29] LINHAO LUO, Yuan-Fang Li, Reza Haf, and Shirui Pan. 2023. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *The Twelfth International Conference on Learning Representations*.

[30] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* 3, 29 (2018).

[31] Péter Mernyei and Cătălina Cangea. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901* (2020).

[32] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

[33] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155* (2022).

[34] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge & Data Engineering* 01 (2024), 1–20.

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[36] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024).

[37] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3973–3983. https://aclanthology.org/D19-1410/

[38] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. http://networkrepository.com

[39] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. 2020. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. ACM, 3125–3132.

[40] Andrew Sherman. 2018. *Mergers and Acquisitions from A to Z*. HarperCollins Leadership.

[41] Qiaoyu Tan, Ninghao Liu, Xiao Huang, Soo-Hyun Choi, Li Li, Rui Chen, and Xia Hu. 2023. S2GAE: Self-Supervised Graph Autoencoders are Generalizable Learners with Graph Masking. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 787–795.

[42] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239* (2022).

[43] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv*

preprint arXiv:2302.13971 (2023).
[44] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJXMpikCZ
[45] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rklz9iAcKQ
[46] Lequn Wang and Thorsten Joachims. 2023. Uncertainty Quantification for Fairness in Two-Stage Recommender Systems. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 940–948.
[47] Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4281–4294.
[48] Ziming Wang, Jun Chen, and Haopeng Chen. 2021. EGAT: Edge-featured graph attention network. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part I 30*. Springer, 253–264.
[49] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. *arXiv preprint arXiv:2303.17564* (2023).
[50] Thomas Yue and Chi Chung Au. 2023. GPTQuant's Conversational AI: Simplifying Investment Research for All. *Available at SSRN 4380516* (2023).
[51] Claudia Zeisberger, Michael Prahl, and Bowen White. 2017. *Private Equity in Action: Case Studies from Developed and Emerging Markets*. Wiley.
[52] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2019. GraphSAINT: Graph Sampling Based Inductive Learning Method. In *International Conference on Learning Representations*.
[53] Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks. In *Proceedings of the ACM on Web Conference 2024*. 1003–1014.
[54] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2021. GreaseLM: Graph REASoning Enhanced Language Models. In *International Conference on Learning Representations*.
[55] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An Empirical Study of Graph Contrastive Learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung (Eds.), Vol. 1. Curran.
[56] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. *arXiv preprint arXiv:2006.04131* (2020).

A Additional Information of CompanyKG (V2)

A.1 Errors, Noises and Redundancies

Each ET in the graph has a varying degree of incompleteness. In most cases, quantifying the degree of completeness is challenging, especially for edge types that rely on third-party information. Consequently, we recommend using all ETs together when training graph learning algorithms on CompanyKG to mitigate this issue. Furthermore, both node features and edge weights can be prone to noise: (1) The node features are based on company description embeddings, and the quality of these embeddings is influenced by the completeness and relevance of the raw textual descriptions. (2) The statistics used to calculate edge weights may be somewhat inaccurate. For instance, inaccuracies may arise in determining people's affiliations when deriving edge weights for ET8.

Because CompanyKG integrates multiple data sources, we established an entity resolution system to merge information from different data sources. During that merging process, two types of errors may emerge: (1) incorrectly merging two different companies into a single graph node and (2) representing the same company as different nodes. The core of the entity resolution system is a trained DNN classifier. We have an entity matching error report UI where users can report the duplicated entities. Over the past year, we received 95 cases among 15,465 active companies (the

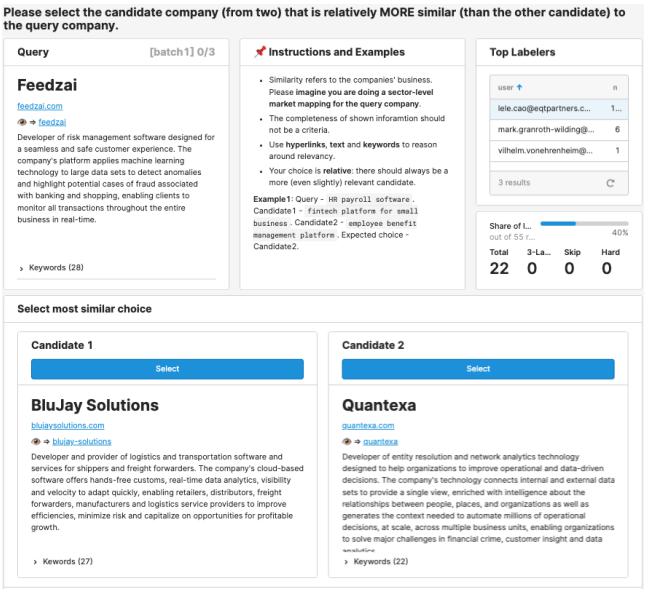


Figure 8: Screenshot of the web application for labeling SR task.

companies that any user has worked on), representing an empirical duplication ratio of merely 0.6%.

A.2 Collection, Annotation and Software

The integration of multi-source information into unified Motherbrain company entities was performed by the Data Engineers at EQT Motherbrain. The construction of the CompanyKG dataset from these entities was carried out by the authors of this paper. The individuals involved in creating the four evaluation datasets (SP, SR, CR, and EP) are detailed below:

- SP: Two EQT Motherbrain employees verified the labels.
- SR: Ten EQT Motherbrain employees (mainly data scientists and machine learning engineers) labeled the first 119 questions. Subsequently, approximately 15 experienced labelers hired by Appen (https://appen.com) labeled 2,400 carefully curated questions at a cost of about 18,000 EUR. See Appx. D-C-2 in [8] for more details.
- CR: Four experienced EQT employees manually extracted data from PE deal materials.
- EP: The first author created SQL queries to capture future edges in ET2, 3, 4, 5, 8, 9, 14, and 15. Additionally, he manually selected 36 similarity-informative collections for ET3.

Since the "raw" data isn't publicly available, we do not release the software for preprocessing or cleaning it. For the SP task, EQT's proprietary platform, Motherbrain, was utilized to acquire labels. For the SR task, we developed a labeling web application using Retool (https://retool.com), as depicted in Figure 8.

B Experimental Settings and Hyper-Parameters

GraphSAGE was trained on a Linux machine with 8 vCPUs, 52GB RAM, 1024GB SSD Disk, and one Nvidia Tesla P100 GPU. The training utilized a mini-batch methodology, achieved by sampling a restricted number of neighboring nodes, and was limited to a maximum of two epochs due to the large scale of the dataset. We

Approach / Model			SP AUC	SR Acc%	CR Recall@50	CR Recall@100	CR Recall@200	CR Recall@500	CR Recall@1k	CR Recall@2k	CR Recall@5k	CR Recall@10k
Graph heuristics		USP	N/A*	59.61	18.27	29.48	40.38	57.62	72.46	75.56	**78.74**	83.08
		WSP (Weighted SP)	N/A*	61.16	**43.69**	**56.03**	**62.72**	**68.67**	**73.10**	**75.58**	77.62	79.37
		Neighbors	0.6229†	56.52	22.25	31.84	46.50	60.16	66.81	67.19	67.19	67.19
		W. Neighbors	0.6020	56.65	43.50	54.65	60.86	65.86	67.19	67.19	67.19	67.19
Embedding proximity		mSBERT (512-dim)	0.8060	67.14	12.96	18.24	23.03	31.10	41.43	47.71	55.84	63.49
		ADA2 (1536-dim)	0.7450	67.20	14.09	21.69	28.41	37.61	48.05	56.13	66.00	71.18
		SimCSE (768-dim)	0.7188	61.69	7.66	8.90	11.94	14.93	18.55	23.48	33.21	41.79
		PAUSE (32-dim)	0.7542	65.19	6.84	9.62	13.11	20.84	31.96	39.60	51.45	65.92
N-shot Prompt.		ChatGPT-3.5	0.7501†	66.73	30.06	31.10	31.91	N/A‡	N/A‡	N/A‡	N/A‡	N/A‡
Knowledge Graph: GNN-based Methods (initiated with different node embeddings)	mSBERT	GraphSAGE	0.7415 ±0.0102	62.03 ±3.11	10.80 ±3.61	13.04 ±2.15	17.63 ±1.60	25.29 ±1.82	33.68 ±0.82	43.29 ±2.40	53.82 ±1.11	64.30 ±1.69
		GRACE	0.7243 ±0.0233	59.36 ±0.64	2.68 ±1.82	4.64 ±1.51	8.03 ±1.34	13.10 ±0.29	20.03 ±0.28	30.61 ±1.32	43.98 ±2.21	53.46 ±4.58
		MVGRL	0.7208 ±0.0336	58.29 ±0.74	2.17 ±1.03	3.56 ±0.33	5.83 ±0.86	10.65 ±1.25	15.52 ±3.61	23.37 ±2.79	35.96 ±2.18	44.37 ±0.47
		GraphMAE	0.7981 ±0.0063	**67.61** ±0.11	20.88 ±0.46	27.83 ±0.39	36.48 ±0.53	50.27 ±0.26	56.21 ±0.37	64.43 ±0.76	76.06 ±0.40	84.69 ±0.40
		eGraphMAE	0.7963 ±0.0030	67.52 ±0.03	18.44 ±0.21	23.79 ±0.22	32.47 ±0.20	42.68 ±0.55	50.82 ±0.41	61.39 ±0.18	70.73 ±0.88	79.69 ±1.10
	ADA2	GraphSAGE	0.7422 ±0.0202	62.90 ±2.25	10.12 ±3.91	11.93 ±0.96	17.51 ±2.44	24.85 ±0.92	32.87 ±1.90	43.26 ±1.74	54.98 ±1.92	63.13 ±1.42
		GRACE	0.7548 ±0.0264	58.20 ±0.47	3.09 ±0.59	4.83 ±0.78	7.60 ±1.20	12.88 ±2.79	21.40 ±2.45	31.63 ±5.07	44.33 ±5.58	54.71 ±4.24
		MVGRL	0.6638 ±0.0557	55.85 ±0.88	1.00 ±0.35	1.77 ±0.24	3.27 ±0.90	7.11 ±1.56	10.85 ±3.33	15.42 ±3.11	24.04 ±6.59	33.38 ±7.50
		GraphMAE	0.8132 ±0.0053	65.10 ±0.06	24.14 ±0.30	29.15 ±0.26	35.06 ±0.36	46.24 ±0.52	58.21 ±0.67	67.53 ±0.36	75.61 ±0.21	**88.32** ±1.20
		eGraphMAE	OOM!	OOM!	OOM!	OOM!	OOM!	OOM!	OOM!	OOM!	OOM!	OOM!
	SimCSE	GraphSAGE	0.7390 ±0.0095	59.90 ±2.26	10.26 ±0.92	14.07 ±3.06	17.47 ±1.24	25.71 ±0.86	38.13 ±2.25	45.84 ±1.87	54.80 ±0.70	64.77 ±0.94
		GRACE	0.7149 ±0.0524	57.26 ±1.38	0.39 ±0.57	1.07 ±0.86	2.25 ±1.65	4.34 ±3.31	6.93 ±4.87	11.77 ±6.69	19.25 ±9.51	29.53 ±11.09
		MVGRL	0.7180 ±0.0600	55.82 ±0.69	0.79 ±0.10	1.20 ±0.45	2.14 ±1.03	4.95 ±1.57	8.36 ±2.31	14.11 ±3.77	21.22 ±4.06	29.33 ±4.68
		GraphMAE	0.8108 ±0.0066	65.46 ±0.05	19.67 ±1.14	26.61 ±0.63	33.24 ±2.20	44.32 ±1.16	53.06 ±2.32	63.45 ±2.67	76.45 ±1.77	84.39 ±0.95
		eGraphMAE	**0.8253** ±0.0089	66.09 ±0.38	18.53 ±0.19	26.33 ±0.17	33.42 ±0.91	44.39 ±0.58	51.50 ±1.02	63.71 ±0.76	74.69 ±0.52	83.81 ±0.24
	PAUSE	GraphSAGE	0.7421 ±0.0121	60.09 ±1.83	5.79 ±1.24	8.02 ±2.03	12.20 ±1.77	17.22 ±1.81	24.82 ±1.70	31.98 ±0.89	42.84 ±7.07	55.22 ±2.47
		GRACE	0.6698 ±0.0135	58.15 ±0.78	0.66 ±0.36	1.31 ±0.36	2.07 ±0.48	5.44 ±0.71	9.58 ±0.80	15.40 ±1.51	26.27 ±4.46	36.47 ±3.97
		MVGRL	0.6843 ±0.0280	55.60 ±1.18	0.47 ±0.35	1.25 ±0.56	2.94 ±0.77	6.39 ±1.76	11.19 ±2.67	17.29 ±2.72	26.68 ±0.79	35.52 ±1.83
		GraphMAE	0.7727 ±0.0113	64.81 ±0.40	10.16 ±1.13	12.51 ±0.62	14.87 ±0.43	19.59 ±0.46	25.41 ±1.32	31.90 ±0.12	42.85 ±1.13	54.33 ±2.75
		eGraphMAE	0.7742 ±0.0068	65.17 ±0.95	11.62 ±0.33	13.95 ±1.50	15.42 ±0.97	19.28 ±0.44	26.88 ±0.53	37.49 ±1.28	49.92 ±0.75	62.53 ±1.47

* We omit SP evaluation for the path-based graph heuristics, since, whilst they are able to rank companies by similarity, there is no obvious way to obtain a 0-1 similarity score for a pair of companies from them.
† To account for the binary nature of the SP prompts answered by ChatGPT, we report accuracy (Acc.) instead of AUC.
‡ As ChatGPT's answer on the CR task is not limited to the companies in CompanyKG and is not mapped to any specific company/node IDs, we conducted a manual examination of the top-K responses and counted the number of entries that match the ground truth competitors. As a result, when the value of K exceeds 200, it becomes overly tedious to calculate hit rate for ChatGPT.

Table 4: The performance of the popular and state-of-the-art baselines on three evaluation tasks of CompanyKG: SP, SR and CR which are compared with AUC, Accuracy (Acc.) and Recall@K respectively. Best results are in bold. "OOM!" means out-of-memory.

adopted the "SAGEConv" implementation from the DGL (https://www.dgl.ai) library with a GCN aggregator.

GRACE and MVGRL were trained on a Linux machine with 8 vCPUs, 52GB RAM, 300GB SSD Disk, and one Nvidia V100 GPU. Both models employed a mini-batch scheme, loading neighbors of nodes sampled in each mini-batch to a depth sufficient to compute the full loss function for the sampled nodes. We used Adam optimization with a learning rate of 0.1, betas of 0.9 and 0.999, and no weight decay, training each model for 100 epochs.

GraphMAE was trained on a Linux machine with 12 vCPUs, 85GB RAM, 1024GB SSD Disk, and one Nvidia Tesla A100 GPU. Given that GraphMAE is a full-batch algorithm, we employed a randomized multi-scale mini-batch approach to accommodate the entire graph in GPU memory. Training was performed for up to 1,000 epochs with an early stopping mechanism. The learning rate was decayed at the end of each epoch using a cosine decay schedule. eGraphMAE, which requires significantly more memory than GraphMAE, was conducted on a Linux machine with 24 vCPUs, 170GB RAM, a 1024GB SSD Disk, and one Nvidia Tesla A100 GPU. We used the same optimal set of hyper-parameter values as GraphMAE, with an additional hyper-parameter for the number of dimensions in the output edge embeddings, set to 32.

	Hyper-params	Searched	Selected hyper-params			
			mSBERT	ADA2	SimCSE	PAUSE
GraphSAGE	# neighbors	6, 12	12	**12**	12	12
	# GNN layers	2, 3	2	**2**	2	2
	batch size	2^{10}, 2^{12}	2^{12}	$\mathbf{2^{12}}$	2^{12}	2^{12}
	epochs	1, 2	1	**1**	1	2
	dropout rate	.1, .3	.3	**.3**	.3	.1
	learning rate	.001, .0001	.0001	**.0001**	.001	.001
	embedding dim	2^5, 2^6, 2^7	2^6	$\mathbf{2^7}$	2^7	2^5
GRACE	# neighbors	5, 10	**5**	10	10	10
	# GNN layers	2, 3	**3**	2	3	3
	hidden dim	8, 16, 32	**16**	32	8	16
MVGRL	# neighbors	5, 10	10	5	10	**5**
	# GNN layers	2, 3	2	3	2	**3**
	hidden dim	8, 16, 32	16	16	8	**16**
(e)GraphMAE	# GNN layers	2, 3	**2**	2	2	2
	# attention heads	4, 8	**8**	4	4	4
	node mask rate	.1, .3, .5, .7	**.3**	.3	.7	.5
	dropout rate*	.1, .3	**.3**	.1	.1	.1
	learning rate	.001, .0005	**.001**	.0005	.001	.001
	edge drop rate	.1, .3, .5, .7	**.1**	.5	.5	.3
	embedding dim	2^6, 2^7, 2^8, 2^9	$\mathbf{2^8}$	2^9	2^7	2^7

* It refers to the dropout rate of the attention dropout layer; the dropout rate of the feature dropout layers is always set to 0.1 higher than the attention dropout rate, up to a maximum of 1.0.

Table 5: Searched and selected (bold) hyper-params for GraphSAGE, GRACE, MVGRL and (e)GraphMAE over different node feature types.

The grid-searched and selected hyper-parameters for each model are gathered in Table 5. For implementation details, please refer to https://github.com/llcresearch/CompanyKG2