DOMAIN ADAPTATION & MULTI-HOP REASONING

DOMAIN-SPECIFIC ADAPTATION AND MULTI-HOP REASONING IN CHEMISTRY AND BIOMEDICINE

By MOHAMMAD KHODADAD,

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree Master of Applied Science

McMaster University

MASTER OF APPLIED SCIENCE (2025)

Hamilton, Ontario, Canada (School of Computational Science and Engineering)

TITLE:	Domain-Specific Adaptation and Multi-Hop Reasoning in Chemistry and Biomedicine
AUTHOR:	Mohammad Khodadad
SUPERVISOR:	Dr. Hamidreza Mahyar
NUMBER OF PAGES:	xix, 143

Lay Abstract

Large language models often excel at general text but struggle with specialized scientific language. This thesis addresses this challenge with three main contributions. First, it introduces ChemTEB and MedTEB: two benchmark collections of 35 chemistry and 51 medical tasks, respectively, covering a range of text-analysis challenges. Second, it presents MedTE, a new 768-dimensional embedding model trained to better understand biomedical language, which achieves leading results on MedTEB. Third, it describes GraphRAG, an automated system that builds chemical knowledge graphs from research preprints and generates complex, multi-step questions to test reasoning. Our experiments reveal significant gaps in current models' grasp of scientific text, with accuracy falling below 50% on multi-step chemistry questions. All benchmarks, code, and models are publicly released to advance research in specialized NLP.

Abstract

Large language models (LLMs) and embedding techniques have transformed general-purpose NLP, but their performance degrades on specialized scientific texts. In this thesis, we make three contributions to bridge this gap. First, we introduce two large-scale benchmark suites: ChemTEB, comprising 35 tasks on chemical corpora drawn from PubChem, CoconutDB, Safety Data Sheets, and Wikipedia; and MedTEB, comprising 51 medical tasks spanning EHR notes, PubMed abstracts, and clinical question–answer sets. Both cover classification, clustering, pair classification, retrieval, and bitext mining. Second, we propose MedTE, a 768-dimensional embedding model fine-tuned via self-supervised contrastive learning on an extensive biomedical corpus, which achieves state-of-the-art performance on MedTEB. Third, we develop GraphRAG, an automated pipeline that constructs chemical knowledge graphs from ChemRxiv preprints and generates multi-hop questions to assess compositional reasoning. Through rigorous evaluation, we show that ChemTEB reveals critical weaknesses in current chemical embeddings and that even with perfect context, LLMs achieve under 50% accuracy on multi-hop chemistry question answering. We release all benchmarks, code, and models to foster further research in domain adaptation and compositional reasoning for specialized NLP applications.

To my wife,

Ghazale

Acknowledgements

I am deeply thankful to my supervisor, **Dr. Hamidreza Mahyar**, for offering me this opportunity and for his steadfast mentorship throughout my Master's studies. Portions of this work were conducted during my internship at BASF; I also appreciate **Soheila Samiee** for her valuable insights and guidance on applying AI in chemistry. I am grateful to McMaster University for cultivating a rigorous and supportive academic environment that has fostered both my intellectual development and research progress. Special acknowledgement goes to the Digital Research Alliance of Canada and BASF Canada for supplying critical computational infrastructure, this research would not have been feasible without their support. This work was funded in part by **MITACS** (grant IT32409).

Table of Contents

List of Figures

List of Tables

Notation, Definitions, and Abbreviations

Notation

$\mathcal{D}$	A dataset, typically a set of input–output pairs used for training or evaluation.
x	A text sample or document (e.g., a sentence or paragraph).
y	A label associated with input x in classification tasks.
z	Embedding vector in $\mathbb{R}^d$ produced by an embedding model: $z = f(x)$.
f	Embedding function or model transforming text x into vector z.
$\cos(z_i, z_j)$	Cosine similarity between embeddings z_i and z_j.
τ	Similarity threshold used in pair classification tasks.
k	Number of clusters in clustering tasks (Mini-Batch k-means).
$nDCG@k$	Normalized Discounted Cumulative Gain at cutoff k, a retrieval metric.

$\mathcal{G} = (V, E)$	Knowledge graph with node set V and edge set E.
K	Number of hops in a multi-hop question (path length minus one).
RRF_{score}	Reciprocal Rank Fusion score used for aggregating model rankings.
F_1	F_1 score, harmonic mean of precision and recall.

Definitions

Embedding A dense, low-dimensional vector representation of text that captures semantic similarity.

Contrastive Learning

A training objective that pulls semantically similar pairs together and pushes dissimilar pairs apart in embedding space.

Domain-Adaptive Pretraining (DAPT)

Continued pretraining of a language model on domain-specific text to improve domain relevance.

Parameter-Efficient Fine-Tuning (PEFT)

Techniques (e.g., LoRA, adapters) that update only a small subset of model parameters.

Retrieval-Augmented Generation (RAG)

A method combining retrieval of external documents with language model generation for knowledge-intensive tasks.

Knowledge Graph (KG)

A graph structure representing entities as nodes and their relationships as edges.

Multi-Hop Question Answering

A QA task where answering requires reasoning across multiple linked facts or graph hops.

Normalized Discounted Cumulative Gain (nDCG)

A measure of ranking quality that accounts for the position of relevant items.

Abbreviations

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CoT	Chain-of-Thought prompting
DAPT	Domain-Adaptive Pretraining
DQN	Deep Q-Network (not used but example)
EHR	Electronic Health Record
GPU	Graphics Processing Unit
ICD	International Classification of Diseases
KG	Knowledge Graph

LLM	Large Language Model
LoRA	Low-Rank Adaptation
MIMIC-IV	Medical Information Mart for Intensive Care IV
MLM	Masked Language Modeling
MTEB	Massive Text Embedding Benchmark
NER	Named Entity Recognition
nDCG	Normalized Discounted Cumulative Gain
NLP	Natural Language Processing
PEFT	Parameter-Efficient Fine-Tuning
PMC	PubMed Central
PubMed	Public MEDLINE database of biomedical literature
RAG	Retrieval-Augmented Generation
RRF	Reciprocal Rank Fusion
SDS	Safety Data Sheet
SMILES	Simplified Molecular Input Line Entry System
SMOTE	Synthetic Minority Over-sampling Technique (not used)
UMLS	Unified Medical Language System
URL	Uniform Resource Locator

VEV	Vertex-edge-vertex (graph notation, not used)
Wiki	Wikipedia

Declaration of Academic Achievement

I hereby declare that all academic work presented in this thesis reflects my own contributions and that, for each included project or publication, I served as a primary collaborator. Chapters 3 to 6 are based on collaborative work with co-authors; at the end of each chapter, a detailed breakdown of contributions, the division of work among authors, and the resources used is provided. The associated papers on which these chapters are based are also listed therein.

- M. Khodadad, A. Shiraee Kasmaee, M. Astaraki, and H. Mahyar, "Towards Domain Specification of Embedding Models in Medicine," *arXiv*, Jul. 2025, arXiv:2507.19407. [Online]. Available: `https://arxiv.org/abs/2507.19407` [80]

- A. Shiraee Kasmaee, M. Khodadad, M. A. Saloot, N. Sherck, S. Dokas, S. Samiee, and H. Mahyar, "ChemTEB: Chemical Text Embedding Benchmark, an Overview of Embedding Models Performance & Efficiency on a Specific Domain," *arXiv preprint*, arXiv:2412.00532, 2024. [Online]. Available: `https://arxiv.org/abs/2412.00532` [78]

- M. Khodadad, A. Shiraee Kasmaee, M. Astaraki, N. Sherck, H. Mahyar, and S. Samiee, "Evaluating Multi-Hop Reasoning in Large Language Models: A Chemistry-Centric Case Study," *arXiv preprint*, arXiv:2504.16414, 2025. [Online]. Available: `https:`

//arxiv.org/abs/2504.16414 [79]

Chapter 1

Introduction

1.1 Motivation

Natural language processing advanced rapidly with contextual embeddings and transfer learning (e.g., ELMo [138], ULMFiT [62]) and then scaled dramatically via the Transformer architecture [182]. Pre-training became a dominant paradigm with models like GPT [142] and BERT [45], which were subsequently refined (e.g., RoBERTa [103], XLNet [209]) and complemented by empirical scaling laws that showed predictable gains from larger models and data [77]. This foundation-model approach, exemplified by unified frameworks such as T5 [146], enabled broad adaptability to downstream tasks with minimal task-specific tuning [20]. Few-shot and zero-shot capabilities emerged with large models: GPT-2 demonstrated fluent closed- and open-domain generation [144], GPT-3 revealed surprising generalization from prompt conditioning alone [26], and later work both scaled (PaLM [36]) and democratized (OPT [212], BLOOM [90], LLaMA [178]) the paradigm. Multimodal reasoning appeared with GPT-4 [128], and conversational fine-tuning produced ChatGPT, whose rapid

adoption underscored real-world impact [129].

Despite these powerful general-language capabilities, large language models (LLMs) struggle when deployed on specialized domains. Most are trained on broad, publicly available corpora, web scrapes, Wikipedia, and public-domain books, which cover everyday expressions and common sense but diverge significantly from texts in fields such as biomedicine, law, and finance. These domains contain unique terminology, formal structures, and dense factual content, creating a distribution shift that degrades accuracy and consistency when generic models are applied without adaptation.

Empirically, this gap is evident and addressable. Vanilla BERT underperforms on biomedical text mining because its pre-training data mismatch the vocabulary and style of biomedical literature; continuing its pre-training on PubMed abstracts and PMC full texts yields BioBERT, which substantially improves named-entity recognition and domain question answering [91]. Similarly, FinBERT, BERT further pre-trained on financial documents, better captures sentiment nuances in earnings reports than general models [7], and SciBERT, trained on scholarly publications with a domain-derived vocabulary, outperforms vanilla BERT on scientific NLP tasks by modeling domain syntax and terminology more faithfully [15]. ClinicalBERT, adapted to hospital notes and electronic health records, improves clinical entity extraction and relation classification over generic counterparts [6]. Gururangan et al. [53] systematically confirm that a second pre-training phase on in-domain text consistently boosts downstream performance across varied fields, showing that wide-coverage pre-training alone cannot substitute for domain focus.

Deeper challenges exacerbate this mismatch:

1. **Vocabulary and tokenization fragmentation.** Fixed subword tokenizers

can break rare or technical terms into incoherent pieces, erasing their semantic integrity (e.g., "cyclooxygenase" being split into meaningless fragments under standard BPE). Domain-specific vocabularies, as in SciBERT's SciVocab, preserve such terminology and yield more coherent embeddings [15, 23].

2. **Hallucination and factual unreliability.** LLMs optimize next-token likelihood, which can produce plausible but incorrect content, harmless in open domains but dangerous in high-stakes settings like medicine or law. Models with limited exposure to rare disease literature, for instance, may invent clinical details that sound credible but lack validity. Survey work identifies hallucination as a core barrier to real-world adoption of foundation models [156], and even domain-tuned systems remain imperfect: Med-PaLM, despite surpassing medical exam thresholds, still makes clinically unacceptable reasoning errors [107], and GPT-4, without explicit medical fine-tuning, can generate overconfident falsehoods. These phenomena show that high benchmark scores do not guarantee dependable understanding [124].

3. **Brittle multi-step reasoning.** Because LLMs are trained for next-token prediction rather than deductive correctness, they can assemble superficially coherent reasoning chains that rely on shortcuts, skip essential inferences, or misapply operations. Their logical quality degrades under minor input perturbations, masking fragility behind fluency [192, 41]. This undermines trust in applications requiring rigorous inference.

To mitigate these limitations, a layered solution emerges. First, *domain-adaptive*

pre-training sharpens internal representations by exposing models to in-domain corpora, aligning them closer to downstream tasks. Second, *retrieval-augmented generation* (RAG) grounds model outputs in explicit source texts, reducing hallucination by conditioning generation on retrieved evidence; grounding has been shown to substantially lower factual error rates compared to closed-book generation [95, 202, 217, 4]. Third, high-quality *embedding models* are central to any retrieval pipeline: they map queries and documents into a shared semantic space. Examples include Sentence-BERT [149], large-scale contrastive embeddings like E5 [188], instruction-tuned multi-vector retrievers such as M^{3}Embed [32], and domain-specific variants (e.g., BioSentVec, FinSent, Mol2Vec). Benchmarks such as MTEB empirically demonstrate that tailored embeddings drive state-of-the-art retrieval and downstream performance across diverse tasks [116].

Together, these components form a coherent workflow: (1) adapt the model's internal representations via domain-adaptive pre-training; (2) ground generation in retrieved, in-domain evidence through RAG; and (3) enhance retrieval and downstream tasks with specialized embeddings. This multilayered pipeline offers a path toward dependable, accurate LLM performance in high-stakes scientific domains, although significant research and engineering challenges remain.

In summary, LLMs in specialized settings expose four critical limitations:

1. Training–target mismatch without domain adaptation.
2. Vocabulary/tokenization issues that fragment rare or technical concepts.
3. Hallucination and factual unreliability in expert domains.
4. Brittle multi-step reasoning that can obscure superficial fluency.

Addressing these gaps requires systems that are not merely large, but domain-aware, grounded, and logically robust.

We advance domain-aware NLP through three core contributions:

- **MedTE Medical Embedding Model:** A self-supervised, contrastively trained embedding model for medical text, enhanced with hard negative sampling and trained on diverse biomedical sources (e.g., PubMed abstracts, PMC full texts, clinical notes), which surpasses existing general and medical embeddings on the MedTEB suite, primarily targeting limitations (1) and (2).
- **ChemTEB and MedTEB Benchmark Suites:** Two standardized evaluation suites in chemistry and medicine covering over eighty tasks (classification, clustering, retrieval, pair classification, bitext mining), with fixed splits, metrics, and baselines that enable reliable comparison and diagnosis of embedding performance, supporting analysis of (1), (2), and (4).
- **GraphQA Multi-Hop Reasoning Pipeline:** An end-to-end system that constructs knowledge graphs from scientific text, synthesizes grounded multi-hop question sets, and evaluates LLM reasoning with and without retrieval augmentation, directly addressing limitations (3) and (4).

1.1.1 Scope and Domains

This work addresses NLP in two specialized scientific domains: *chemistry* and *medicine*. Chemical literature abounds with complex nomenclature, reaction descriptions, and structural formulas rendered as plain text, while medical sources range from clinician notes to research papers and patient-generated narratives, each with domain-specific

terminology and abbreviations. We limit our study to *text-based* English data, omitting multimodal materials (e.g., chemical diagrams, radiology scans) and other fields such as legal or financial texts.

Our investigation comprises four projects: (1) creating a specialized medical embedding model, (2) designing a medical text embedding benchmark, (3) building a chemical text embedding benchmark, and (4) evaluating LLMs on multi-hop question answering over knowledge graphs. Rather than training new general-purpose architectures from scratch, we focus on adapting, assessing, and extending pre-trained models to satisfy chemistry- and medicine-specific requirements.

1.1.2 Problem Statement and Research Questions

Although LLMs perform strongly on broad-coverage benchmarks, they frequently falter on specialized corpora due to vocabulary mismatch, domain drift, and the hallucination of unsupported information. This thesis addresses the overarching question:

> *How can we systematically adapt and evaluate pre-trained language models to deliver reliable embeddings and reasoning mechanisms in chemical and medical contexts?*

We pursue four research questions:

1. **Domain-Adaptive Training:** In what ways can additional pre-training or precise fine-tuning boost embedding performance on medical text while preserving general-language capabilities?

2. **Benchmarking:** Which embedding models excel or fall short on targeted chemical and medical text tasks, and which evaluation framework best highlights

these differences?

3. **Reasoning Evaluation:** How do different LLMs perform in multi-hop question answering over chemical knowledge graphs?

1.1.3 Contributions

This thesis makes three primary contributions to domain-aware NLP:

- **MedTE Medical Embedding Model:** We develop MedTE, a contrastively trained, self-supervised embedding model built on diverse biomedical corpora, which surpasses existing alternatives on the MedTEB suite.
- **ChemTEB and MedTEB Benchmark Suites:** We introduce two embedding benchmarks, ChemTEB (chemistry) and MedTEB (medicine), that together cover over eighty tasks in classification, clustering, retrieval, pair classification, and bitext mining.
- **GraphQA Multi-Hop Reasoning Pipeline:** We present GraphQA: an end-to-end system that constructs domain knowledge graphs from scientific text, generates multi-hop questions, and benchmarks LLM reasoning with and without retrieval augmentation.

1.1.4 Thesis Organization

The thesis proceeds as follows:

- **Chapter 1: Introduction** Outlines motivations, scope, and the four projects spanning chemistry and medicine.

- **Chapter 2: Literature Review** Surveys text embedding techniques, domain-specialized LLMs (e.g., BioBERT, SciBERT), and knowledge-graph QA methods.
- **Chapter 3: MedTE Model** Describes the design, training procedure, and empirical evaluation of our medical embedding model.
- **Chapter 4: MedTEB Benchmark Suite** Details datasets, tasks, and evaluation results for the medical embedding benchmark.
- **Chapter 5: ChemTEB Benchmark Suite** Presents the chemical embedding benchmark's datasets, tasks, and performance metrics.
- **Chapter 6: GraphQA Pipeline** Explains knowledge-graph construction, question set generation, and multi-hop QA experiments for LLM assessment.
- **Chapter 7: Conclusion** Synthesizes findings, discusses limitations, and outlines future research directions in domain-aware NLP.

‘

Chapter 2

Literature Review

2.1 Embedding Techniques

2.1.1 Static Embeddings

Word embeddings map discrete word types to fixed vectors in a continuous space, capturing semantic and syntactic regularities such that analogous word pairs (e.g., *king–queen*) exhibit linear relationships [179, 112]. Grounded in the distributional hypothesis, "you shall know a word by the company it keeps" [46], early approaches like Latent Semantic Analysis (LSA) apply singular value decomposition to word–document co-occurrence matrices to produce dense vectors [42].

Neural predictive models then transformed the field. Word2Vec introduced two architectures: Continuous Bag-of-Words (CBOW) and Skip-Gram, that learn embeddings by predicting a target word from its context or vice versa, using negative sampling for efficiency [113, 112]. These embeddings encode both semantic similarity and analogical structure (e.g., $\mathbf{v}(king) - \mathbf{v}(man) + \mathbf{v}(woman) \approx \mathbf{v}(queen)$) [111].

Count-based global methods like GloVe factorize a corpus-wide co-occurrence matrix, optimizing so that the dot product of two word vectors approximates the logarithm of their co-occurrence count [134]. Empirical comparisons find that, when carefully tuned, predictive (Word2Vec) and count-based (GloVe) embeddings achieve similar quality by capturing comparable distributional information [13, 93].

A key limitation of basic static embeddings is handling out-of-vocabulary words and morphological variants. FastText addresses this by representing each word as a bag of character n-grams, summing subword vectors to form a word embedding, thereby generating representations for rare or unseen words [19].

Another drawback is polysemy: a single vector must aggregate all senses of a word. Multi-sense static embeddings attempt to learn multiple prototype vectors per word by clustering contexts [67] or via non-parametric extensions of Skip-Gram that allocate a variable number of sense-specific embeddings per word [123]. While improving the representation of homonymous words, these approaches still assign a fixed set of vectors per word type, without adapting to sentence-level context.

Static embeddings thus offer simplicity, low computational cost (just a lookup table), and ease of training on modest corpora. However, their context-independence limits them when confronting polysemy and nuanced language phenomena, paving the way for contextual techniques.

2.1.2 Contextual Embeddings

Contextual embeddings compute a word's vector dynamically from its sentence or document context, enabling disambiguation of polysemous words and capturing fine-grained usage nuances [137].

2.1 RNN-based

ELMo (Embeddings from Language Models) derives token representations from internal states of a deep bidirectional LSTM trained with a language modeling objective (predicting both next and previous words) [137]. Each word instance receives a context-sensitive embedding by combining the representations from multiple LSTM layers, yielding substantial gains on tasks like question answering, coreference resolution, and sentiment analysis when integrated into downstream models [137]. On top of that, ULMFiT showed that fine-tuning a pre-trained LSTM language model can yield strong task performance [61];

2.2 Transformer-based

The transformer architecture's self-attention mechanism [181] heralded a new era of contextual embedding. GPT (Generative Pre-Training) employs a unidirectional transformer trained to predict the next token, producing rich left-to-right contextual representations [141]. GPT-2 scaled this approach, demonstrating that larger models trained on more data yield increasingly powerful embeddings and language generation capabilities [143].

BERT (Bidirectional Encoder Representations from Transformers) uses a masked language modeling objective and next-sentence prediction to learn deep bidirectional context representations [44]. By integrating information from both left and right contexts simultaneously, BERT produced token embeddings that, when fine-tuned, achieved state-of-the-art results across benchmarks like GLUE and SQuAD [186, 44].

Subsequent models expanded this paradigm: XLNet introduced a permutation-based autoregressive objective to capture bidirectional context without masking [208]; Flair

combined character-level language models for sequence tagging tasks [2]; and GPT-3, with 175 billion parameters, demonstrated zero- and few-shot learning abilities from its contextual embeddings [27]. By 2020, these transformer-based contextual embeddings underpinned the "foundation models" paradigm, pre-training on massive corpora then fine-tuning for specific tasks [21].

2.3 Specialized Variants

To further tailor contextual embeddings, researchers have developed variants optimized for contrastive objectives, domain specificity, and specialized applications:

• **Contrastive-focused pretraining** General Text Embedding (GTE) combines unsupervised contrastive pretraining on large unlabeled corpora with supervised fine-tuning on labeled pairs (e.g., NLI, QA) to produce universally transferable embeddings [98]. Nomic-Embed blends masked language modeling and contrastive learning over very long contexts (up to 8,192 tokens) using efficient architectures (FlashAttention, DeepSpeed), matching or exceeding larger models on semantic and retrieval benchmarks despite a modest 137M parameter count [125].

• **General-domain variants** SciBERT is a BERT variant pretrained on a corpus of scientific publications (biology, chemistry), capturing domain-specific vocabulary and style to outperform BERT on scholarly tasks [15]. Sentence-BERT (SBERT) fine-tunes siamese/triplet BERT networks on NLI and semantic similarity data to produce sentence embeddings directly comparable via cosine similarity, improving clustering and STS performance [149]. E5 models employ large-scale weakly supervised contrastive learning on billions of paired texts (e.g., page titles and contents) to learn

robust universal embeddings for retrieval and classification [188]. Instruction-tuned and multilingual embeddings like M^{3}Embed extend embedding length (up to 8192 tokens) and unify dense, sparse, and multi-vector retrieval in one model by incorporating task instructions during pretraining [32].

• **Clinical/Biomedical variants** Med-BERT introduces cross-visit pretraining on structured EHR sequences, learning from longitudinal patient records to improve clinical outcome prediction [148]. ClinicalBERT continues BERT pretraining on MIMIC-III clinical notes to capture medical jargon and narrative style for tasks such as entity recognition [69]. BioBERT trains on PubMed abstracts and PMC full texts, boosting performance on biomedical text-mining benchmarks [91]. ExBERT integrates an out-of-vocabulary module for domain-specific terms [172], while GatorTron, an 8.9B-parameter transformer trained on 90B de-identified clinical notes and literature, sets new standards on clinical NLI, STS, and QA tasks at the cost of heavy computation [205].

• **Clinical/Biomedical And Contrastive-focused** Further biomedical refinements apply contrastive and self-supervised tuning: BioSimCSE adapts SimCSE to biomedical sentences for robust sentence embeddings [76]; Abro et al. perform self-supervised contrastive tuning on ClinicalBioBERT to refine clinical segment representations [1]; Min et al. use ChatGPT to generate paraphrase pairs for contrastive learning [114]; NoteContrast aligns clinical notes with ICD-10 codes in a contrastive framework [75]; MICOL leverages metadata for hierarchical contrastive learning to improve zero-shot multi-label classification [216]; MedEmbed employs triplet-based

contrastive pretraining on PubMed with hard negatives for fine-grained discrimination [11]; and BioLORD-2023 combines UMLS knowledge-graph augmentation, contrastive learning, and self-distillation to advance clinical semantic similarity and retrieval benchmarks [150].

2.1.3 Static vs Contextual

Static embeddings provide lightweight, efficient word representations learned from relatively small corpora, but each word type has a single vector that conflates all senses and ignores sentence-level context. Contextual embeddings dynamically tailor representations to each occurrence, disambiguating polysemy and encoding rich syntactic and semantic information, which yields superior performance on most language understanding tasks [137, 44].

Nonetheless, contextual models demand substantially more data, parameters, and computing resources. They often require GPU-accelerated inference and pretraining on billions of tokens [21]. Static embeddings remain appealing in low-resource or real-time settings due to their simplicity, interpretability (via linear algebra analyses), and modest computational footprint. Hybrid approaches that extract static vectors from contextual models (e.g., by averaging contextualized representations) or initialize contextual models with static embeddings reflect the continuum between these paradigms [50].

Ultimately, the choice hinges on task requirements and resource constraints: for lightweight, downstream tasks or scarce-data domains, static (or small contextual) embeddings may suffice; for advanced language understanding, contextual embeddings, especially specialized variants, offer clear advantages in accuracy, flexibility,

and domain adaptation.

2.2 Large-Scale Embedding Benchmarks

Large-scale benchmarks are widely used to evaluate text embedding models across diverse tasks and domains. For example, SentEval provides a toolkit for assessing sentence representations on classification, entailment, and similarity tasks [38]. GLUE (and SuperGLUE) aggregate tasks like natural language inference, sentiment analysis, and question answering to test general-purpose model performance [186, 185]. Indeed, GLUE's nine tasks include sentiment (SST-2), paraphrase detection (MRPC, QQP), NLI (MNLI, RTE), and QA (QNLI), among others [186]. SuperGLUE was introduced when top systems surpassed human performance on GLUE [185]. The STS Benchmark (STS-B) provides an explicit evaluation of semantic textual similarity [28], complementing classification benchmarks. For retrieval evaluation, BEIR comprises 18 retrieval datasets from varied domains (e.g. scientific, news, forums) to test zero-shot generalization [175]. For example, BEIR includes tasks such as COVID-19 question answering, debate forum retrieval (ArguAna), and scientific fact verification (SciFact) [175]. The long-running TREC conferences have produced many IR and QA collections for evaluating embedding-based search. The Massive Text Embedding Benchmark (MTEB) spans eight tasks over 58 datasets in 112 languages, representing the most comprehensive evaluation of text embeddings to date [118]. Through MTEB, models are tested on tasks from semantic search to clustering and classification, ensuring wide coverage of embedding applications.

Multilingual benchmarks extend these ideas across languages. XTREME evaluates multilingual encoders on 40 languages and nine tasks, highlighting cross-lingual

transfer challenges [65]. XGLUE offers 11 cross-lingual tasks covering both understanding and generation, enabling pretraining on large multilingual corpora [99]. For example, XGLUE includes tasks such as cross-lingual sentiment analysis and question answering, as well as machine translation. CLUE provides the first large-scale Chinese NLU benchmark, with nine tasks (e.g. classification, reading comprehension) on Chinese text [203]. These multilingual suites often adapt GLUE-like tasks to various languages or use parallel-data tasks, testing whether embeddings capture language-agnostic semantics.

Another important dimension is inference and paraphrase. Large NLI corpora like MultiNLI (433k pairs across diverse genres) gauge models' ability to generalize inference beyond narrow domains [197]. MultiNLI's size notably exceeds earlier NLI datasets, providing a more challenging benchmark. PAWS provides 108k paraphrase vs. non-paraphrase pairs with high lexical overlap [215], explicitly testing models' sensitivity to word order and context. For example, "flights from New York to Florida" vs. "flights from Florida to New York" share all words but reverse meaning; PAWS includes such examples to expose weaknesses of bag-of-words models [215]. These datasets show that embeddings must capture subtle semantic cues rather than simple word overlap.

Domain-specific embedding benchmarks have also emerged. In healthcare, Clinia's heMTEB extends the MTEB framework to biomedical data, adding tasks like the CURE dataset of clinical passage retrieval across ten specialties (in English, French, Spanish) [37]. (Soffer et al. also proposed a scalable clinical embedding evaluation framework, highlighting the need for specialized benchmarks.) These efforts focus on medical terminology, multi-disciplinary records, and cross-domain evaluation, since

embeddings trained on general text often underperform in this domain.

Within biomedical NLP, specialized semantic similarity and inference datasets are used. BIOSSES provides 100 biomedical sentence pairs manually annotated for similarity [168]. MedSTS contains 1,068 clinical sentence pairs annotated on a 0–5 similarity scale [191]. MedNLI contains nearly 15k premise–hypothesis pairs from clinical notes, annotated by doctors for entailment/neutral/contradiction [153]. These resources complement general benchmarks by focusing on medical concept similarity and reasoning. For example, evaluation on MTEB shows that no single embedding model dominates across all tasks [118], indicating that different benchmarks can favor different embedding methods. In practice, researchers often report combined scores (e.g. average accuracy on GLUE or MTEB) to summarize overall performance, while also examining individual task results to understand model tradeoffs. For instance, models specialized for sentence similarity may excel on STS but not on retrieval, whereas dense retrievers might lag on classification tasks. BEIR results even highlight that traditional sparse methods like BM25 remain strong baselines in zero-shot retrieval [175], underscoring the diversity of evaluation needs.

In summary, embedding evaluation now spans general NLU suites (GLUE, SuperGLUE, SentEval), semantic similarity benchmarks (STS-B, BIOSSES, MedSTS), retrieval collections (BEIR, TREC), multilingual benchmarks (XTREME, XGLUE, CLUE), and domain-specific datasets (heMTEB/CURE, MedNLI). Each benchmark contributes to understanding model performance in its context, and together they guide researchers toward more robust and versatile embedding models.

2.3 Sequence Generation and Seq2Seq Models

2.3.1 RNN-Based Generative Models

Recurrent neural networks (RNNs) have long been a foundational model for sequence modeling in natural language processing. Classic RNNs maintain a hidden state that is recurrently updated, allowing information to persist across sequence positions. Early RNNs suffered from vanishing and exploding gradients, making long-range dependencies difficult to learn in practice [17]. Gated variants addressed this limitation: Long Short-Term Memory (LSTM) networks introduced input, output, and forget gates plus an explicit memory cell to overcome vanishing gradients [59], while Gated Recurrent Units (GRUs) simplified the gating mechanism with comparable performance [34].

In the sequence-to-sequence setting, Sutskever et al. mapped an input sequence to a fixed-length vector with an LSTM encoder and then generated an output sequence with an LSTM decoder, demonstrating strong machine translation performance [171]. Bahdanau et al. improved this by introducing a soft-attention mechanism that lets the decoder attend dynamically to encoder hidden states, alleviating the fixed-vector bottleneck for long or complex inputs [9].

Chemical Sequence Generation Chemical structures can be represented as SMILES strings, making them amenable to RNN-based generation. LSTM models trained on large SMILES corpora learn chemical syntax and can generate novel compounds with desired properties [161, 126]. These generative SMILES models facilitate de novo drug design by proposing plausible bioactive molecules. RNNs have also been used for molecular property prediction (QSAR) by treating SMILES as character sequences

and learning continuous embeddings for toxicity, solubility, or activity forecasting; combined with graph neural networks, they achieve competitive results on benchmark datasets [204].

Clinical Sequence Modeling In healthcare, RNNs encode time-series data (e.g. lab measurements, vital signs) and unstructured clinical notes for tasks such as risk prediction and outcome forecasting. The RETAIN model, an attention-augmented LSTM, predicts heart failure risk by attending to critical visits and features in EHR sequences [35]. More broadly, surveys report that LSTMs and GRUs excel at modeling longitudinal patient data for early detection of deterioration and disease progression [164]. RNNs are also applied to clinical text analysis, using bi-directional LSTMs to label sequences of words in doctor's notes for diagnosis coding or concept extraction.

2.3.2 Transformer-Based Seq2Seq Models

The Transformer architecture replaces recurrence with multi-head self-attention, enabling parallel sequence modeling and robust long-range dependency capture [181]. In its encoder–decoder form, the Transformer uses self-attention in both encoder and decoder, plus cross-attention in the decoder to attend to encoder outputs, achieving state-of-the-art translation performance.

GPT-style decoder-only Transformers are pretrained autoregressively and excel at generative tasks with few-shot prompting. GPT-3, with 175 billion parameters, demonstrates strong performance across translation, question answering, and summarization without task-specific fine-tuning [27]. Encoder–decoder models like T5 cast all tasks as text-to-text generation and fine-tune on labeled data, achieving top results on summarization and QA benchmarks [145].

Chemical Reaction Prediction & Molecule Generation Transformer-based seq2seq models frame reaction prediction as SMILES-to-SMILES translation. The Molecular Transformer predicts reaction products and yields directly from reactant SMILES, achieving over 90% top-1 accuracy without hand-crafted rules [159]. Pre-trained variants such as Chemformer (BART-based) further improve performance on retrosynthesis and optimization tasks [71]. Autoregressive molecular generators like MolGPT produce valid novel SMILES and can be conditioned on scaffolds or properties [8].

Clinical Report Summarization & Note Generation Encoder–decoder Transformers (e.g. BART, T5) fine-tuned on clinical corpora summarize radiology reports, discharge notes, and physician–patient dialogues with human-level quality [180]. Domain-specific LLMs such as BioGPT (pretrained on PubMed abstracts) excel at biomedical QA and relation extraction [106], while Med-PaLM 2 (PaLM 2 fine-tuned on medical data) achieves over 85% accuracy on USMLE-style questions [166]. These models also aid in drafting clinical notes from patient data with enhanced factuality and coherence [97].

2.4 RAG Frameworks (e.g. REALM, RAG)

Retrieval-augmented generation (RAG) frameworks extend large language models by incorporating an explicit retrieval component. For example, Guu et al. [55] introduce *REALM*, which learns to retrieve relevant text (e.g. Wikipedia passages) during

masked language model pre-training. REALM's model attends to these retrieved documents via backpropagation through the retrieval step, integrating external knowledge into its representations. Similarly, Lewis et al. [95] propose RAG, a fine-tuned seq2seq generator coupled with a neural retriever. At inference, RAG fetches the top-k documents from a large corpus (often indexed with FAISS) and conditions the generator on the retrieved passages to answer a query. These systems typically use dense embedding retrieval (e.g. DPR) or traditional sparse search (BM25) to find relevant context. By grounding generation in retrieved evidence, RAG models inject up-to-date information without full retraining and mitigate hallucinations, since the output is constrained by explicit sources [202]. Indeed, Lewis et al. [95] report that RAG achieved state-of-the-art accuracy on multiple open-domain QA benchmarks, far outperforming closed-book LMs.

2.4.1 Chemistry: retrieving reaction protocols or spectral data

In chemistry, RAG can exploit specialized knowledge sources (reaction databases, literature, spectral libraries) to improve predictions and textual output. Zhong et al. [217] introduce *ChemRAG-Bench*, a benchmark integrating heterogeneous chemical knowledge (scientific articles, PubChem entries, PubMed abstracts, textbooks, Wikipedia, etc.) for RAG evaluation. They find that augmenting LLMs with retrieval yields large gains (17% average improvement over direct inference) on diverse chemistry tasks. For example, one study retrieves similar molecules (using chemical fingerprints) and their reference mass spectra to guide prediction: the MARASON

model uses retrieved structures and spectra to simulate spectra, boosting top-1 accuracy from 19% to 28% [189]. In practice, chemical RAG could query resources like the Open Reaction Database or patent literature to retrieve detailed experimental protocols, or match unknown spectral data against libraries (e.g. NIST) to produce informed analyses.

2.4.2 Medicine: retrieving PubMed abstracts, EHR-grounded RAG

In medicine, RAG systems ground generation in authoritative biomedical text. Xiong et al. [202] introduce the MIRAGE benchmark and MedRAG toolkit, evaluating medical QA systems that retrieve from corpora like PubMed abstracts, clinical guidelines, and textbooks. They observe that RAG provides large gains: on biomedical QA datasets (e.g. PubMedQA, MMLU-Med), adding retrieval from PubMed yields 1–18% absolute improvement over closed-book models. Notably, they find PubMed retrieval boosts performance across all tasks, given its broad coverage. Domain-specific retrievers (e.g. MedCPT) or multi-retriever ensembles can further improve accuracy. RAG also aids clinical information tasks: Alkhalaf et al. [4] demonstrate that augmenting an LLM with retrieval of related clinical notes raises EHR summarization accuracy from 93% to 99%. By retrieving relevant abstracts or patient-specific documents at inference, medical RAG models can incorporate up-to-date evidence and reduce hallucination in clinical QA and information extraction.

2.5 Knowledge Graphs and Graph-Based Reasoning

Knowledge graphs (KGs) encode structured knowledge as triples (h, r, t), where each head h and tail t is an entity and r is a relation, all defined under an explicit ontology. By representing knowledge in this way, KGs enable semantic integration of heterogeneous data sources and support principled graph-based reasoning over the encoded domain. Ontologies (e.g., OWL) provide the schema, classes, and relations, while the KG itself instantiates that schema with real-world data. Graph-based reasoning then consists of inferring new facts or answering queries by traversing and applying algorithms (symbolic or learned) over this graph structure.

2.5.1 KG Construction & Embeddings (TransE, GAT)

Construction of KGs can be manual (curation from databases) or automatic via information extraction. Recent work leverages large language models (LLMs) to extract entities and relations from scientific text. For example, Langer *et al.* automatically construct a chemical entity-role KG by fine-tuning transformer-based NER models (e.g., BERT) to recognize chemical entities and then using a second LLM (LLaMA-2) to verify relation assertions, mapping extracted triples to ChEBI identifiers and outputting RDF-formatted KG data [88]. Such pipelines dramatically accelerate graph population compared to manual curation.

Once constructed, KGs are often embedded into vector spaces to support tasks like link prediction. TransE [22] represents entities and relations as vectors $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ and enforces $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ for true triples. This simple translation property effectively

models many relation types and remains a standard baseline. Graph neural networks (GNNs) further generalize embeddings by learning over multi-relational graphs. In particular, Graph Attention Networks (GATs) [183] apply self-attention over node neighborhoods, enabling nodes to weigh the importance of each neighbor when updating their representation. In KG contexts, relational GNN variants (e.g., R-GCN) and attention-based GATs provide powerful end-to-end learning frameworks for link prediction and node classification.

2.5.2 General Multi-Hop QA over KGs (HotpotQA, QAngaroo)

Multi-hop question answering (QA) tests a model's ability to combine multiple facts to answer queries. HotpotQA [206] is a large-scale dataset of over 110,000 Wikipedia questions that explicitly require multi-document reasoning: each question is annotated with two supporting paragraphs that must be combined. HotpotQA also includes explainable supporting facts to encourage models to justify each inference step. Similarly, the QAngaroo benchmark [193] comprises WikiHop (open-domain Wikipedia QA) and MedHop (PubMed QA) subsets. WikiHop requires retrieving evidence chains across Wikipedia articles to answer a query, while MedHop focuses on drug–protein interactions across scientific abstracts. Although these datasets are text-based, their multi-hop structure mirrors path-finding in a KG, identifying a chain of connected facts that lead to the answer. KG-aware QA systems typically retrieve candidate passages, construct a subgraph of entities and relations, and perform either symbolic graph traversal or neural graph reasoning (e.g., attention over the subg raph) to infer answers.

2.5.3 Chemistry: ChEBI, PubChem Graphs & Reaction Networks

In chemistry, KGs draw upon established ontologies and massive compound databases. The Chemical Entities of Biological Interest (ChEBI) ontology provides a curated hierarchy of $\sim 2 \times 10^5$ biologically relevant molecules [47]. PubChem, by contrast, is not an ontology but a vast repository of over 100 million chemical substance records, available in RDF for KG integration. Reaction ontologies (e.g., RXNO) and specialized frameworks like OntoRXN formalize chemical reaction networks as graphs of species and transformation steps [47]. OntoRXN treats each reaction mechanism as a graph of molecular species connected by reaction-step nodes, enabling integration with computational chemistry data (e.g., energy profiles).

Chemical KGs often integrate multiple sources: ontologies (ChEBI, RXNO), substance databases (PubChem), and literature. The FORUM KG links chemicals to diseases and genes by combining ChEBI, ChemOnt, and PubChem data with co-occurrence and ontological inference [88]. More recently, LLM-driven pipelines extract chemical entities and roles directly from papers, automatically mapping them to ChEBI IDs and constructing large-scale chemical KGs [88]. These AI-assisted methods promise to extend KG coverage far beyond manual curation.

2.5.4 Medicine: UMLS, SNOMED CT, and Clinical KGs

Medical KGs are rooted in standardized terminologies. SNOMED CT [40] is the largest clinical ontology, containing hundreds of thousands of medical concepts (diseases, procedures, findings) organized in a polyhierarchy. The Unified Medical Language System (UMLS) [101] integrates over 200 biomedical vocabularies into a metathesaurus, linking millions of concepts with semantic relations. These terminologies themselves form KGs of concepts and relations. Chang *et al.* demonstrate that embedding the SNOMED-CT subgraph using relational graph models outperforms text-only embeddings for biomedical similarity tasks [30].

Beyond terminologies, application-specific clinical KGs integrate patient data, molecular interactions, and disease ontologies. The SPOKE network [115], for instance, links genes, drugs, and diseases to support translational research. PrimeKG [29] aggregates dozens of biomedical databases into a unified graph connecting phenotypes, pathways, drugs, and genes. Clinical NLP systems leverage UMLS entity linking to anchor unstructured text in a KG context, enabling downstream tasks such as cohort discovery and decision support.

2.5.5 LLM-Assisted KG Extraction and Reasoning

Recent advances apply LLMs both to KG construction and to graph-based reasoning. LLMs facilitate entity and relation extraction from unstructured text at scale, mapping to ontology IDs and populating KGs automatically [88]. On the reasoning side, experiments show that LLMs can be utilized to form multi-hop questions chained in order and to decompose complex queries into sequential retrieval and inference steps, thereby enhancing their accuracy on domain-specific RAG benchmarks [195].

Overall, the synergy of curated ontologies (ChEBI, UMLS), large-scale graph databases (PubChem), and LLM-driven extraction is rapidly advancing domain-specific KG generation. Embedding methods (TransE, GAT) and KG-aware QA benchmarks (HotpotQA, QAngaroo) underpin the graph-based reasoning architectures that bridge symbolic knowledge and neural language models, offering a path toward robust, explainable LLM applications in chemistry and medicine.

2.6 Agentic QA & Multi-Hop Reasoning Agents

2.6.1 Tool-Using LLM Agents (ReAct, Toolformer)

Agentic language model agents can interleave reasoning steps with explicit tool calls to external APIs or databases. In the ReAct framework proposed by Yao et al. [210], an LLM is prompted to produce alternating `Thought` (reasoning) and `Action` (tool use) steps. This interleaving enables multi-step reasoning that remains grounded in verifiable intermediate observations. On interactive decision-making benchmarks (e.g. ALFWorld, WebShop), ReAct-style prompting substantially outperformed imitation and reinforcement learning baselines by allowing the model to reason and act in tandem [210]. Similarly, Toolformer (Schick et al.) fine-tunes an LLM with self-supervised signals to invoke external tools when needed [158]. Toolformer's training process automatically labels places in the model's own outputs where calling a tool (such as a calculator, search engine, or knowledge-base API) would be beneficial, and trains the model to incorporate those tool calls into its generation [158]. By learning when and how to delegate sub-tasks to specialized tools, Toolformer achieves significantly better zero-shot performance on arithmetic, factual lookup, and reasoning

tasks without degrading the base model's core language generation abilities [**?**].

2.6.2 GraphQA: Chaining Retrieval and Reasoning

Graph-based QA agents extend retrieval-augmented generation by structuring the retrieved evidence as an explicit graph and then chaining the reasoning over this graph. For example, HopRAG (Liu et al.) constructs a graph of passages during retrieval, where edges represent semantic relatedness, and uses an LLM to iteratively propose sub-queries and traverse the graph in multiple hops [102]. At each hop, top-k new passages are retrieved and added to the graph, and the LLM's reasoning helps identify which node to explore next. This retrieve-reason cycle continues until the query is answered, enabling a form of logical multi-hop exploration beyond flat list retrieval [102]. GraphRAG (Han et al.) takes a related approach by integrating a dense retriever with a graph-based reranker that uses self-attention over the retrieved subgraph to select the most coherent chain of evidence [56]. The agent builds a graph of candidate passages and employs a graph neural network or attention mechanism to find an optimal multi-hop path that connects the question to the answer [56]. LLMs can also translate natural language questions into structured graph queries , for instance, GMeLLo (Chen et al.) maps questions into sets of RDF triples and then executes multi-hop SPARQL queries on a knowledge graph to retrieve the answer [33]. By leveraging the precision of graph-structured data, GMeLLo handles sequential logic and evolving knowledge bases in multi-hop QA settings [33]. These GraphQA methods demonstrate that explicitly modeling a graph of retrieved knowledge and reasoning step-by-step over that graph yields more accurate and interpretable multi-hop answers than flat retrieval alone. This was foreshadowed by Lewis et al. [95], who

showed that even in early RAG systems, chaining retrieval with reasoning reduces hallucinations and leads to better answer correctness [95].

2.6.3 Domain-Specific QA Agents in Chemistry and Medicine

In specialized domains like chemistry and medicine, agentic QA systems combine LLMs with domain-specific toolkits to achieve expert-level performance. ChemCrow (Bran et al.) is an LLM-based chemistry agent that integrates GPT-4 with 18 expert chemistry tools (for tasks such as molecule search, reaction prediction, spectrometry, safety checking, etc.) [24]. The ChemCrow agent is prompted to plan complex workflows by sequentially calling the appropriate tools for each sub-problem in tasks spanning organic synthesis, drug discovery, and materials design [24] . This approach allowed ChemCrow to autonomously plan and execute the synthesis of various compounds (e.g. an insect repellent and organocatalysts) by combining GPT-4's reasoning with the precise outputs of chemistry software tools [24]. Building on this idea, ChemAgent (Wu et al.) scales up to an arsenal of 137 chemistry APIs and employs a Hierarchical Evolutionary Monte Carlo Tree Search (HE-MCTS) to plan multi-step tool-using strategies [198]. In ChemAgent, the LLM chooses which tool to use at each step (e.g. a reaction predictor vs. a docking simulator) and in what sequence, with the MCTS algorithm exploring different tool sequences to maximize the expected success of the overall plan [198]. Through reinforcement learning fine-tuning, ChemAgent learned to navigate this large tool set efficiently, yielding state-of-the-art results on benchmarks for reaction prediction and molecular design by virtue of combining LLM reasoning with rigorous domain-specific computations [198]. In the medical domain, LLM agents are augmented with clinical knowledge bases and

specialized reasoning strategies to tackle complex clinical questions. Med-PaLM 2 (Singhal et al.) is an example of a medical QA agent built on Google's PaLM-2 LLM and fine-tuned for healthcare applications [167]. Med-PaLM 2 incorporates a "chain-of-retrieval" prompting strategy [167]: it iteratively retrieves relevant information from medical databases like PubMed and clinical guidelines at each step of reasoning, feeding those facts back into the model before it generates the next part of the answer. This retrieval-augmented approach dramatically improved Med-PaLM's performance on U.S. Medical Licensing Exam (USMLE)-style questions , Med-PaLM 2 scored about 19% higher than a comparable closed-book model on MedQA, approaching or exceeding state-of-the-art accuracy on multiple-choice clinical benchmarks [167]. Likewise, multi-agent frameworks have been explored to reflect the collaborative nature of clinical reasoning. The MedAgentsBench suite (X. Tang et al.) was introduced as a benchmark to evaluate multi-step medical reasoning with multiple agents [173]. Experiments on MedAgentsBench found that combining an LLM with specialized medical tools and reasoning agents (for example, separate "specialist" agents for diagnosis, treatment planning, etc.) can outperform a single large model prompting approach by up to 25% on complex diagnostic and planning problems [173]. These domain-specific QA agents underscore the importance of deeply integrating LLMs with curated toolsets and structured knowledge bases: by leveraging external domain expertise (through tools or retrieval) and guiding the model's reasoning process, they achieve far more reliable and expert-level performance in high-stakes fields like chemistry and medicine than would be possible with an LLM alone.

2.7 Domain Adaptation & Specialization

2.7.1 Domain-Adaptive Pretraining (DAPT).

Continuing the pretraining phase on unlabeled domain-specific corpora is a proven way to specialize a large language model to a target domain. This domain-adaptive pretraining (DAPT) technique, first highlighted by Gururangan et al. (2020) [52], involves taking a general English model and further training it (masked-language or autoregressive) on domain text (e.g. biomedical papers or chemical patents) so that the model "sees" more in-domain vocabulary and patterns. Many studies have found that DAPT yields significant in-domain performance gains. For example, Alhmoudi et al. (2025) observed that continuing to pretrain a chemical language model on a specialized molecules corpus led to better performance, especially when the target domain differs greatly from generic web text [3]. Similarly, in materials science, Huang and Cole (2025) demonstrated a cost-efficient DAPT on optoelectronics literature that cut pretraining compute by over 80% while still matching or exceeding the baseline model's accuracy [66]. However, DAPT is not a free lunch: Huang and Cole also point out that many scientific fields lack the massive text corpora that are typically used to train a transformer from scratch, making extensive in-domain pretraining impractical [66]. In summary, DAPT can inject crucial domain knowledge (as exemplified by models like SciBERT or BioBERT) and often improves downstream accuracy, but its success hinges on having sufficient domain-specific data and compute resources [52] [3] [66].

2.7.2 Vocabulary & Tokenization Adaptations.

A key challenge in specialist domains is handling terms not seen in generic corpora. Standard subword tokenizers (WordPiece/BPE) often over-segment technical terms. For example, Balde et al. (2025) report that open-domain LLM tokenizers fragment many medical terms into multiple pieces, yielding high out-of-vocabulary (OOV) rates on biomedical text [12]. One strategy to mitigate this is expanding or rebuilding the model's vocabulary using domain data. In biomedicine, for instance, PubMedBERT was pretrained with a fresh WordPiece vocabulary derived from PubMed abstracts, in contrast to BioBERT which reused BERT's original lexicon. Domain-specific tokenizers can better reflect the morphology of technical terms, although models can sometimes be surprisingly robust to imperfect segmentation [54]. In chemistry, researchers have even introduced alternative token representations: for example, SELFIES is a chemically-aware string encoding that guarantees valid molecules [84], but adopting it requires building a new tokenizer and pretraining on a large SELFIES corpus [3]. In practice, even modest vocabulary augmentation can help. Balde et al. showed that adding common medical jargon to an LLM's vocabulary improved clinical text summarization quality on inputs with many OOV terms [12]. Likewise, expanding the vocabulary of BERT/RoBERTa with domain-specific terms has yielded gains on various classification and generation tasks [54, 12]. Overall, tailoring tokenization, either by adding key domain tokens or retraining the subword segmenter on domain text , tends to reduce OOV fragmentation and better preserve important terms in medicine and chemistry, albeit at the cost of retraining embeddings for the new tokens.

2.7.3 Fine-Tuning Strategies: LoRA/PEFT in Chem/Med.

Once a base model (possibly after DAPT) is obtained, it often needs task-specific fine-tuning on limited labeled data. Training all billions of parameters is expensive, so parameter-efficient fine-tuning (PEFT) methods are popular. Approaches like LoRA (Low-Rank Adaptation) and related PEFT schemes freeze the original weights and insert a small number of trainable adapter parameters. For example, Balde et al. note that fine-tuning large LLMs via quantized LoRA (QLoRA) has become common practice, since full end-to-end updates of huge models are infeasible in many cases [12]. In the medical domain, Sukeda et al. (2024) demonstrate that LoRA-based instruction tuning , i.e. fine-tuning a model on medical Q&A examples using a frozen backbone plus low-rank adapters , can indeed inject domain-specific knowledge into an LLM, with larger base models seeing the biggest gains [170]. Similarly, He et al. (2025) propose a "parameter-sensitive" LoRA fine-tuning method that efficiently adapts an LLM to specialized Q&A tasks with limited expert data, showing substantial accuracy improvements on both medical and legal question-answering benchmarks [58]. In practice, lightweight adapters and QLoRA have been used in both chemistry and biomedicine to conserve resources. For instance, chemical LLM systems like ChemCrow use small domain-specific adapter modules when tuning models for tasks like synthesis planning. Instruction tuning (providing domain-relevant exemplars or prompts) is another complementary strategy , often combined with LoRA-based fine-tuning , to align models with expert tasks [170]. In summary, PEFT methods such as LoRA and QLoRA enable affordable specialization: they focus the training on learning domain-critical patterns from scarce labeled data while leaving the vast majority of model weights untouched [170] [58].

2.7.4 Challenges: OOVs, Hallucinations, Data Scarcity.

Despite these adaptation efforts, domain-specific LMs still face persistent issues. OOV terms remain problematic: even cutting-edge LLMs will split uncommon chemical or medical names into many subword pieces, which can distort meaning. Balde et al. (2025) confirm this, reporting that various model tokenizers all produce much higher fragmentation for medical text than for general-domain news text [12]. Hallucination is another serious risk: models may confidently generate incorrect or physically impossible outputs. In chemistry, this issue surfaces as "molecular hallucinations" – proposing invalid or implausible molecules – which researchers have attempted to curb via reinforcement learning or by applying rank-based losses that penalize chemically invalid generations [57]. In medicine, hallucinated content can lead to dangerously false diagnoses or treatments, so strategies like retrieval augmentation and human feedback (e.g. reinforcement learning from human feedback, RLHF) are employed to improve factuality [130] [57]. Finally, data scarcity is a fundamental challenge in both the chemistry and medical domains. Specialized scientific corpora are typically much smaller (and have far fewer labeled examples) than the web-scale text used to train general-purpose LLMs. Huang and Cole note that their materials science text collection was orders of magnitude smaller than what's used for full-scale pretraining, necessitating the kind of "cost-efficient" DAPT approach they employed [66]. Similarly, Han et al. (2024) emphasize that the limited availability of high-quality chemistry data – and the fact that chemical knowledge is spread across multiple modalities – creates distinct hurdles for LLM development [57]. With few expert-annotated examples, even fine-tuning runs the risk of overfitting. These constraints

mean that domain-specialized LMs often must rely on techniques like data augmentation, retrieval of external knowledge, or cross-domain transfer learning to achieve reasonable performance.

2.8 Evaluation Metrics & Benchmarks

8.1 General Metrics. For supervised classification tasks, *accuracy* (the fraction of correct predictions) is the simplest evaluation metric. However, when classes are imbalanced, metrics such as *precision*, *recall*, and the harmonic mean (F_1 *score*) are preferred [132, 140]. In unsupervised clustering, the *V-measure*, the harmonic mean of homogeneity and completeness, quantifies cluster quality via conditional entropy [154]. Alternatively, *purity* measures the extent to which each cluster contains items from a single class [108].

In text generation tasks such as machine translation and summarization, automatic metrics compare generated text to references. *BLEU* computes n-gram precision with a brevity penalty [132], while *ROUGE* emphasizes n-gram recall and longest common subsequence overlap [100]. *METEOR* aligns words using synonyms and stems, combining precision and recall with a fragmentation penalty [89]. Recent metrics like *BERTScore* use contextual embeddings to semantically match tokens, often correlating better with human judgments [213]. Typical evaluations report classification metrics (accuracy, F_1), clustering metrics (V-measure, purity), and generation metrics (BLEU, ROUGE, METEOR, BERTScore) as appropriate.

8.2 Domain-Specific Metrics: chemical similarity, clinical accuracy. In chemistry, evaluating generated molecules or reaction predictions requires specialized measures. *Tanimoto similarity* (binary Jaccard index on molecular fingerprints) is

standard for assessing compound similarity [10]. Reaction prediction models are assessed via *top-k accuracy* (correct products among top-k predictions) and *round-trip accuracy*, which verifies that predicted reactants yield the target product through a forward reaction model [160]. Additional metrics include *coverage* (fraction of successful predictions) and *diversity* (number of unique valid outputs) [160].

In clinical NLP, evaluation often combines standard metrics with domain-specific checks. Factual consistency can be measured via entailment models trained on MedNLI, ensuring that generated summaries do not contradict reference sentences [152]. Clinical entity coverage evaluates whether key UMLS concepts from references appear in outputs [18]. For QA, accuracy on datasets like PubMedQA and MedQA is reported [73, 167]. These metrics ensure not only fluency but also correctness and completeness of critical domain knowledge.

8.3 Benchmark Suites. Benchmark suites provide standardized evaluation across multiple tasks. In general NLP, *GLUE* [184] and *SuperGLUE* [187] aggregate classification, inference, and reasoning tasks. The *BEIR* benchmark spans 18 zero-shot information retrieval datasets [176]. The *Massive Text Embedding Benchmark (MTEB)* evaluates embeddings on 58 datasets across classification, clustering, retrieval, and generation in 112 languages [117].

In chemistry, *MoleculeNet* offers property prediction tasks on molecular datasets [200]. Generative benchmarks include *MOSES* for SMILES-based molecule generation [139] and *GuacaMol* for de novo design tasks [25]. Community challenges such as *ChemBench* provide leaderboards for reaction prediction and molecular optimization.

In medicine, *BioASQ* provides biomedical QA and summarization with expert-curated

answers [85]. *MedNLI* tests clinical inference [152]. QA benchmarks include *PubMedQA* [73] and *MedQA* [167]. Clinical summarization and entity recognition (e.g., i2b2 challenges) often use ROUGE or F_1 scores alongside domain-specific checklist-based assessments [165]. Together, these benchmarks enable comprehensive evaluation of models across general NLP, chemical informatics, and clinical NLP applications.

2.9 Open Challenges Trends

9.1 Hallucination Mitigation. Large language models frequently generate plausible-sounding but incorrect information, a phenomenon known as "hallucination" [177]. In scientific and clinical domains, such errors can fabricate non-existent drug trials or medical claims, posing serious risks to patient safety [14]. Retrieval-Augmented Generation (RAG) grounds model outputs in external knowledge bases (e.g. PubMed, ChEBI) by concatenating retrieved passages to the input, which dramatically improves factual accuracy in medical question answering from roughly 40% to over 99% [48, 94]. Tool-augmented agents like ReAct interleave reasoning steps with API calls for verification (e.g. chemical validity checks), further reducing hallucinations [210]. Additional strategies include consistency checks, querying the model multiple times and filtering inconsistent responses, and constrained decoding to enforce domain rules such as valid chemical syntax or clinical guideline compliance [177].

9.2 Explainability in Domain Context. Transparent decision-making is vital in chemistry and medicine, where practitioners must understand the rationale behind AI suggestions [68]. Attention visualization techniques display which input tokens or structural elements the model focuses on [68], while feature-importance methods

like LIME [151] and SHAP [105] estimate each input's contribution to the prediction. Probing classifiers, trained on hidden representations, can reveal whether LLMs capture domain-specific concepts (e.g. drug–target interactions) [68]. In clinical summarization, SHAP has been used to highlight critical symptoms and lab values that drive a diagnosis [151], and attention heatmaps guide radiologists through automated report generation [68]. In chemistry, explainable embeddings can pinpoint molecular substructures responsible for predicted properties, aiding medicinal chemists in lead optimization [68].

9.3 Efficient Adaptation (compute/data constraints). Fine-tuning large LLMs for specialized tasks can be prohibitively expensive. Parameter-efficient transfer learning methods, such as adapter layers [60] and Low-Rank Adaptation (LoRA) [63], update only a small fraction of model weights, reducing trainable parameters by orders of magnitude and cutting GPU memory requirements by up to 3×. Data scarcity in medical and chemical domains further complicates adaptation. Domain-aware augmentation, synonym replacement using medical ontologies [31] or SMILES string enumeration for molecules [199], effectively multiplies training examples while preserving domain validity. Combining PEFT with such augmentation has enabled practical fine-tuning of LLMs for tasks ranging from clinical QA to reaction prediction, even on limited hardware [63, 199].

9.4 Future Directions: Multimodal Integration and Causal Reasoning. Future LLMs will increasingly integrate multimodal data. In chemistry, models like ChemVLM combine textual descriptions with molecular graphs or images of chemical diagrams, enabling tasks such as image-to-structure translation and cross-modal molecule design [96]. In healthcare, multimodal models that fuse clinical text with

imaging (X-rays, histopathology) and time-series signals (ECGs) promise richer patient representations [5]. Causal reasoning capabilities are also emerging as critical for robust AI. Causal representation learning aims to embed known cause–effect relations within models, enabling counterfactual queries and confounder handling [81]. Trustworthy AI guidelines such as FUTURE-AI outline principles including fairness, robustness, explainability, and auditability, requirements that next-generation domain-specific LLMs must meet for safe deployment in high-risk environments [92]. Combining multimodal grounding with causal frameworks, under rigorous validation, will chart the path toward reliable, explainable, and effective AI in chemistry and medicine.

Chapter 3

MedTE

3.1 MedTE: A Contrastively Trained Medical Text Embedding Model

Effective medical text embeddings must capture complex clinical terminology, long-tail disease names, and contextual nuances present in electronic health records and biomedical literature. To this end, we build MedTE by fine-tuning a General Text Embedding (GTE) backbone with self-supervised contrastive learning on a richly curated medical corpus. The GTE base model, originally pretrained on broad web and scientific text, provides robust contextual representations [98].

Our contrastive objective draws positive and negative sentence pairs from diverse sources: PubMed abstracts [120], full-text articles from PubMed Central [122], clinical notes in MIMIC-IV [74], trial protocols in ClinicalTrials.gov [121], and preprints from bioRxiv and medRxiv [162, 109]. We leverage in-batch negatives mining to sharpen MedTE's capacity for fine-grained semantic discrimination across medical

subdomains.

By aligning semantically related clinical sentences and separating unrelated ones, MedTE produces dense representations that excel in downstream tasks such as ICD-10 code classification, clinical note retrieval, semantic similarity, and medical question answering. As we demonstrate in Chapter 4, MedTE consistently outperforms both general-purpose and prior medical-domain embeddings on comprehensive benchmarks, underscoring the power of self-supervised contrastive learning for domain-specialized NLP.

3.2 Model Training

Data Preprocessing. For each data source, we construct semantically aligned *positive pairs*, two text fragments conveying the same meaning, according to the rules in Table 3.1. All text is lower-cased and tokenized using the GTE tokenizer (a `bert-base-uncased` variant [43]) to ensure a consistent subword vocabulary.

Training Objective & Architecture. We employ a single-stage, fully unsupervised contrastive learning objective on the GTE-Base transformer. Given a mini-batch $\mathcal{B} = \{(x_i, x_i^+)\}_{i=1}^N$ of N positive pairs, we compute embeddings $z_i = \text{MeanPool}(\text{GTE}(x_i))$ and similarly z_i^+ for the positives. We then minimize the InfoNCE loss [127]:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(\text{sim}(z_i, z_i^+)/\tau\right)}{\sum_{j=1}^{N} \exp\left(\text{sim}(z_i, z_j^+)/\tau\right)},$$

where $\text{sim}(\cdot,\cdot)$ is cosine similarity and τ is a learned temperature. Implicit negatives are provided by all non-matching examples in the batch, eliminating the need for manual negative sampling. To prevent domain shift between sources, each batch is

drawn entirely from a single corpus.

We retain GTE's default mean-pooling over the final hidden states; experiments with CLS-token or max-pooling yielded no performance gains. As the original GTE did not include any MLM phase, and our own trials confirmed that adding masked-language pretraining added cost without benefit, we omit MLM entirely and train end-to-end on the contrastive objective.

Training Setup. We train on the full set of 2,033,800 positive pairs (Table 3.1), sampling each batch of size 1,024 proportionally across sources. Training runs for 8,500 steps, which we found sufficient for convergence. Optimization uses AdamW [104] with weight decay 0.01, a linear warmup over the first 1,000 steps to peak learning rate 2×10^{-5}, followed by cosine decay. Mixed-precision (bfloat16) and gradient checkpointing maximize batch size under our GPU memory constraints, and DeepSpeed [147] accelerates large-batch training. At inference, we L_2-normalize the embeddings so that inner-product search is equivalent to cosine similarity.

Data Source	**Sentence Pair Definition**	**Number of samples**
PubMed	Article *title* ↔ *abstract* sentence #1	572 300
bioRxiv / medRxiv	Pre-print *title* ↔ *abstract* sentence #1	231 400
MIMIC-IV	*History of Present Illness* ↔ *Chief Complaint*	311 400
ClinicalTrials.gov	Study *title* ↔ *detailed description*	378 800
MedMCQA	Exam *question* ↔ *answer explanation*	151 000
MedQA	Exam *question* ↔ *answer explanation*	5 300
MedQuAD	User *question* ↔ authoritative *answer passage*	8 500
TREC-COVID	Search *query* ↔ relevant *passage*	129 200
NF-corpus	Information-need *query* ↔ relevant *document snippet*	3 600
CURE-V1	Clinical *query* ↔ supporting *evidence sentence*	242 300
Total		2 033 800

Table 3.1: Positive-pair construction for self-supervised contrastive learning (rounded).

3.3 Analysis

As a complement to the present work, we developed the MedTEB benchmark to provide a comprehensive and systematic evaluation of embedding models on a variety of medical-domain tasks, including classification, Clustering, Retrieval, and Pair Classification over clinical and biomedical text. In Chapter 4, we show that our proposed model achieves state-of-the-art performance, significantly outperforming all existing baselines across the benchmark. These gains are consistent across different task families and data splits, and persist under rigorous evaluation (e.g., via bootstrapped confidence intervals and statistical significance testing). The training dynamics are further examined in the Appendix, where we present the loss curves for both training and validation.

3.4 Conclusion

MedTE advances medical text embeddings by integrating a GTE backbone [98] with self-supervised contrastive learning to capture the subtle semantics of clinical language. Its 768-dimensional representations, trained on PubMed, PMC, MIMIC-IV, ClinicalTrials.gov, bioRxiv, and medRxiv, achieve state-of-the-art performance in classification, clustering, similarity, and retrieval tasks (Chapter MedTEB). The combination of normalized temperature-scaled contrastive loss [127], in-batch hard negative mining, and parameter-efficient tuning via LoRA [64] ensures both fine-grained discrimination of medical entities and rapid adaptation to new subdomains.

Limitations The current version of MedTE was trained using an unsupervised contrastive learning objective on approximately 2 million samples. While this approach

enabled broad coverage without task-specific labels, the relatively limited dataset size may restrict the model's ability to capture the full diversity of medical language, particularly for rare conditions or specialized subdomains. In addition, the absence of supervised fine-tuning on targeted downstream tasks may limit performance in applications that require precise task-specific optimization.

Future Work Future improvements to MedTE could involve expanding the training corpus to include a substantially larger and more diverse set of medical texts, thereby improving coverage and generalization. Incorporating supervised fine-tuning on curated, task-specific datasets could further enhance performance for applications such as clinical information retrieval, diagnosis support, and question answering. Combining these enhancements with frameworks like Retrieval-Augmented Generation (RAG) and neural rerankers may lead to more accurate and context-aware medical NLP systems capable of integrating the latest literature during inference.

3.5 Attribution

Citation M. Khodadad, A. Shiraee Kasmaee, M. Astaraki, and H. Mahyar, "Towards Domain Specification of Embedding Models in Medicine," *arXiv*, Jul. 2025, arXiv:2507.19407. [Online]. Available: https://arxiv.org/abs/2507.19407 [80]

Contributions This work was developed through collaborative efforts. Mohammad Khodadad contributed to the ideation, preparation of the benchmark, preparation of the models, and writing. Ali Shiraee assisted with ideation and provided consultation. Mahdi Astaraki provided consultation and contributed to writing. Dr. Hamidreza

Mahyar supervised the project and provided guidance throughout.

Resources All training jobs were run on Compute Canada infrastructure, using an *A100 Large* node with 64 CPU cores and 100 GB RAM. The software stack included `Python`, `git`, `openai`, `PyTorch`, and `DeepSpeed`, with package management handled via `pip`. Code is available at `https://zenodo.org/records/16882530` and `https://github.com/MohammadKhodadad/MedTE-dev`. Furthermore, the trained MedTE model (CL15, step 8000) is hosted on Hugging Face and can be accessed at `https://huggingface.co/MohammadKhodadad/MedTE-cl15-step-8000`.

Chapter 4

MedTEB

Medical text embeddings underpin a wide spectrum of healthcare NLP applications, from clinical decision support and biomedical literature retrieval to patient-centric question answering, by transforming unstructured text into dense vector representations. Despite impressive gains in general-domain embedding research, models like BERT [43] struggle to capture the specialized vocabulary, abbreviations, and complex semantics present in medical narratives. Domain-tuned variants such as BioBERT [91], ClinicalBERT [69], and Med-BERT [148] have been proposed to mitigate this, yet their evaluation remains fragmented: many are tested on only a handful of narrow tasks, and often fail to outperform recent general-purpose embedding models like E5 [188] or Sentence-BERT (SBERT) [149] on comparable benchmarks.

Furthermore, existing evaluation suites, whether small-scale clinical similarity sets or broader frameworks like the Massive Text Embedding Benchmark (MTEB) [116], offer limited medical coverage, leaving critical tasks such as diagnosis coding, clinical note retrieval, and patient-centric clustering under-assessed. As a result, it is difficult to determine which embedding approaches truly generalize across the diverse

terminologies and inference patterns encountered in practice.

To address these gaps, we introduce MedTEB, a large-scale benchmark of 51 medical text embedding tasks spanning classification, clustering, pair classification, and retrieval (PubMedQA, clinical note search). We then evaluate a set of medical and non-medical popular embedding models, and analyze their performance. We also evaluate our model, MedTE, which is explained in Chapter 5, and compare it with existing models.

This chapter details the design, data curation, and evaluation protocols of MedTEB, and presents a comprehensive comparison of MedTE and leading baselines. Our findings reveal that MedTE consistently outperforms both prior medical embeddings and general-purpose models across every task, underscoring the necessity of domain-adaptive pretraining and unified benchmarking for robust medical NLP.

4.1 Methodology

4.1.1 Sources

Our corpus combines multiple complementary sources of biomedical knowledge, including peer-reviewed literature, preprint archives, real-world clinical narratives, structured registries, curated encyclopedic content, standardized question–answer datasets, and specialized training corpora. We start with PubMed [120], a comprehensive database of biomedical abstracts, and its full-text counterpart, PubMed Central (PMC) [122], which ground our collection in rigorously reviewed scientific discourse. To capture authentic clinical language, we use MIMIC-IV [74], a de-identified EHR

dataset containing clinical notes, discharge summaries, and structured patient information representative of bedside documentation. Structured trial metadata, objectives, interventions, and eligibility criteria from ClinicalTrials.gov [121] extend our coverage to ongoing and completed research across therapeutic areas. For timely access to emerging findings, we include life-science preprints from bioRxiv [162] and clinically focused preprints from medRxiv [109], both offering rich metadata for fine-grained filtering and trend analysis.

We also incorporate human-curated medical articles from Wikipedia, which provide systematically structured descriptions of diseases, diagnostics, and treatments, linking professional terminology with lay explanations [196]. To assess reasoning and factual recall, we use multiple-choice question datasets: MedMCQA, based on Indian medical entrance exams [131], and MedQA, aligned with US medical licensing assessments [72]. Additionally, the MedQuAD corpus contributes 47,457 question–answer pairs sourced from authoritative providers such as the National Cancer Institute, enhancing coverage of diverse consumer-health inquiries [16].

Collectively, these materials create a comprehensive and well-balanced dataset that merges structured databases, peer-reviewed literature, clinical text, and educational sources. This combination offers a strong basis for developing and testing medical language models in tasks involving retrieval, understanding, and reasoning.

4.1.2 Benchmark Development

In our benchmark, we assembled 51 datasets across four categories, Classification, Clustering, Pair Classification, and Retrieval, to comprehensively evaluate embedding models.

Classification. Given a labeled dataset $\mathcal{D} = (x_i, y_i)i = 1^N$, we fine-tune the embedding model on $\mathcal{D}$train, train a logistic regression classifier on the resulting embeddings, and report macro-averaged F_1 on $\mathcal{D}_{\text{test}}$.

Clustering. For each input x_i, we compute embedding $z_i \in \mathbb{R}^d$ and apply Mini-Batch k-means (batch size 32) to obtain cluster labels $\hat{c}_i$. Clustering quality is measured by V-measure against true labels c_i.

Pair Classification. Each dataset contains pairs $(x_i^{(1)}, x_i^{(2)}, y_i)$ with binary label y_i. We compute four similarity metrics (cosine, Euclidean, Manhattan, dot product), select threshold τ on training data to maximize F_1, and report the best F_1 on test data.

Retrieval. Queries $\mathcal{Q}$ and documents $\mathcal{D}$ are embedded into vectors z_{q_j} and z_{d_k}; we rank documents by cosine similarity $\cos(z_{q_j}, z_{d_k})$ and report nDCG@10 averaged over queries.

Table 4.1 summarizes the presence of data sources across these tasks.

Table 4.1: Presence of Data Sources in MedTEB

Task	MIMIC-IV	PMC	PubMed	Wikipedia	MedQA	MedMCQA	MedQUAD	medRxiv	bioRxiv	Total
Classification	✓	✓	✓	✓	✗	✗	✗	✗	✗	15
Clustering	✓	✓	✗	✓	✗	✗	✗	✗	✗	12
Pair Classification	✓	✗	✗	✗	✓	✓	✗	✓	✗	12
Retrieval	✓	✗	✓	✓	✓	✓	✓	✓	✓	12

These protocols enable a holistic evaluation of embedding quality in both supervised and unsupervised scenarios, capturing semantic grouping, pairwise discrimination, and practical retrieval capabilities across diverse medical text tasks.

4.2 Results

Our evaluation covered a wide range of embedding models, categorized along two main axes: domain focus (general-purpose vs. medical) and training approach (contrastive learning vs. conventional pretraining). Table 4.2 outlines details such as model scale, number of parameters, context length, embedding size, and whether contrastive methods were applied. The non-medical baselines feature transformer architectures like BERT [43], SciBERT [15], and E5 [188], together with lighter-weight Sentence-Transformer versions [149]. The medical group includes ClinicalBERT [69], BioSimCSE [76], MedEmbed [11], as well as our newly introduced MedTE.

Model Name	Size (MB)	Params (M)	Context	Emb. Dim	Cons. Lear.
Non-medical models					
BAAI BGE Base En V1.5	418	109.48	512	768	Yes
AllenAI SciBERT Scivocab Uncased	422	109.92	512	768	No
Google BERT Base Uncased	420	109.48	512	768	No
Intfloat E5 Base	418	109.48	512	768	Yes
Nomic AI Nomic Embed Text V1	522	136.73	819	768	Yes
Nomic AI Nomic Embed Text V1 Unsupervised	522	136.73	512	768	Yes
Sentence-Tfrs All-MiniLM-L6-v2	87	22.71	512	384	Yes
Sentence-Tfrs All-MPNet-Base-v2	418	109.49	512	768	Yes
Thenlper GTE Base	209	109.48	512	768	Yes
Medical models					
BioNLP BlueBERT PubMed MIMIC Uncased L-12 H-768 A-12	420	109.48	512	768	No
Abhinand MedEmbed Base	420	109.48	512	768	Yes
Emily Alsentzer Bio-ClinicalBERT	416	108.31	512	768	No
Kamalkraj BioSimCSE BioLinkBERT Base	413	108.23	512	768	Yes
Malteos SciNCL	419	109.92	512	768	Yes
MedicalAI ClinicalBERT	517	134.73	512	768	No
Microsoft BiomedBERT Base Uncased Abs+Fulltext	420	109.48	512	768	No
MedTE Cl15 Step 8000	438	109.48	512	768	Yes

Table 4.2: Model Specifications for Evaluated Embedding Models, with Domain and Contrastive-Learning Indicators

Table 4.3 presents performance on classification (macro F_1), clustering (V-measure), pair classification (F_1), and retrieval (nDCG@10). Our model, MedTE, achieves the highest scores across all four task families, with an overall average of 0.578, surpassing the next best general model (GTE Base V2) at 0.529 and the top clinical baseline

(MedEmbed) at 0.539.

Model	Classification	Clustering	Pair Cls.	Retrieval	AvgType	AvgAll	EvalTime
Non-medical models							
BAAI Bge Base En V1.5	0.69 ± 0.22	0.28 ± 0.28	0.67 ± 0.15	0.38 ± 0.36	0.505	0.516	102.78
AllenAI Scibert Scivocab Uncased	0.60 ± 0.20	0.06 ± 0.05	0.64 ± 0.12	0.06 ± 0.07	0.341	0.356	99.26
Google BERT Base Uncased	0.58 ± 0.20	0.10 ± 0.11	0.63 ± 0.11	0.08 ± 0.10	0.348	0.362	102.08
Intfloat E5 Base	0.68 ± 0.22	0.24 ± 0.25	0.67 ± 0.15	0.33 ± 0.33	0.479	0.490	102.51
Nomic AI Nomic Embed Text V1	0.69 ± 0.22	0.28 ± 0.28	0.67 ± 0.15	0.38 ± 0.36	0.503	0.511	112.79
Nomic AI Nomic Embed Text V1 Unsupervised	0.69 ± 0.22	0.29 ± 0.21	0.68 ± 0.16	0.40 ± 0.35	0.515	0.525	90.15
Sentence-Transformers All MiniLM L6 V2	0.68 ± 0.21	0.28 ± 0.17	0.66 ± 0.14	0.34 ± 0.33	0.489	0.501	**20.87**
Sentence-Transformers All MPNet Base V2	0.70 ± 0.22	0.30 ± 0.21	0.67 ± 0.15	0.35 ± 0.34	0.502	0.514	54.98
Thenlper GTE Base	0.70 ± 0.22	0.28 ± 0.24	0.69 ± 0.16	0.41 ± 0.35	0.518	0.529	94.98
Medical models							
Abhinand MedEmbed Base	0.69 ± 0.22	0.36 ± 0.28	0.68 ± 0.16	0.39 ± 0.35	0.529	0.539	53.60
BioNLP Bluebert PubMed Mimic Uncased L-12 H-768 A-12	0.62 ± 0.21	0.09 ± 0.11	0.63 ± 0.11	0.07 ± 0.07	0.351	0.367	52.23
EmilyAlsentzer Bio ClinicalBERT	0.59 ± 0.21	0.06 ± 0.06	0.63 ± 0.11	0.05 ± 0.05	0.334	0.349	104.01
Kamalkraj BioSimCSE BioLinkBERT Base	0.64 ± 0.22	0.23 ± 0.24	0.66 ± 0.14	0.27 ± 0.31	0.449	0.460	53.83
Malteos SciNCL	0.69 ± 0.22	0.34 ± 0.26	0.66 ± 0.14	0.30 ± 0.30	0.498	0.509	98.57
MedicalAI ClinicalBERT	0.60 ± 0.21	0.10 ± 0.10	0.63 ± 0.11	0.06 ± 0.06	0.346	0.366	43.07
Microsoft BiomedNLP BiomedBERT Base Uncased Abstract Fulltext	0.61 ± 0.20	0.12 ± 0.17	0.64 ± 0.12	0.13 ± 0.17	0.374	0.388	49.34
MedTE Cl15 Step 8000	**0.72 ± 0.23**	**0.38 ± 0.24**	**0.74 ± 0.17**	**0.45 ± 0.32**	**0.569**	**0.578**	54.63

Table 4.3: Performance of embedding models on various tasks. All values represent the average metric per task.

Classification. MedTE attains an F_1 score of 0.72, outperforming GTE Base (0.70) and MedEmbed (0.69). This translates to significantly fewer misclassifications in downstream clinical decision support tasks.

Clustering. With a V-measure of 0.38, MedTE exceeds MPNet Base V2 (0.30) and MedEmbed (0.36) , demonstrating superior capability in patient subgroup discovery.

Pair Classification. MedTE achieves F_1=0.74 on Pair Classification, improving over GTE Base (0.69) and MedEmbed (0.68).

Retrieval. MedTE achieves nDCG@10=0.45 on retrieval, improving over GTE Base by (0.41) and MedEmbed (0.39). These gains reflect a more semantically coherent embedding space.

Impact of Contrastive Learning. Models without contrastive fine-tuning (BERT, SciBERT, ClinicalBERT) lag significantly across tasks, underscoring that

domain-adaptive contrastive objectives are crucial for capturing nuanced medical semantics .

Overall, MedTEB establishes a new state-of-the-art for medical text embeddings Benchmark, demonstrating the importance of contrastive pretraining, and a comparison of medical and non-medical models.

4.2.1 Per Source Performance

As outlined in the data sources section, MedTEB assesses embeddings across diverse medical text sources. Table 4.4 shows each model's average performance by source. Both Abhinand MedEmbed Base and BAAI Bge Base En V1.5 achieve top scores on PubMed and bioRxiv (0.89–0.90), clearly outperforming SciBERT [15] (PubMed 0.50; bioRxiv 0.18) and BioClinicalBERT [69] (PubMed 0.47; bioRxiv 0.10). MedTE continues this trend, delivering the highest MIMIC-IV score (0.61) and strong results for ClinicalTrials.gov (0.81) and medRxiv (0.74). In contrast, general-purpose embeddings like All-MiniLM L6 v2 and MPNet Base V2 score only 0.51–0.55 on PubMed and 0.64–0.67 on bioRxiv, emphasizing the advantage of domain-specific pretraining.

For the MedMCQA and MedQA tasks, both multi-choice medical question answering, results show a consistent challenge: most models average 0.39, reflecting the inherent difficulty of these benchmarks. Even so, MedTE leads on MedMCQA, likely due to its broad and diverse medical pretraining. On the general-domain Wikipedia task, domain-adapted models show only a slight drop in performance, with MedTE at 0.45 and Thenlper GTE Base at 0.43, outperforming general embeddings such as All-MiniLM L6 v2 (0.41) and BERT Base (0.26). Overall, the per-source breakdown confirms that MedTE outperforms other embeddings on specialized biomedical text

while maintaining competitive results on broader sources.

Figure 4.1 compares model results between MIMIC-IV and Wikipedia tasks, showing that technical clinical notes (MIMIC-IV) are more challenging than encyclopedic text. Nomic Embed Text V1, a general-domain, contrastively trained model, outperforms many dedicated clinical embeddings, although MedTE remains the top performer. The models naturally group into two categories based on whether they were trained with contrastive learning, underscoring its role in capturing domain-specific nuances.

Model	BioRxiv	Clinical Trials	MIMIC-IV	MedMCQA	MedQA	MedQuAD	MedRxiv	PMC	PubMed	Wikipedia
Non-medical models										
BAAI Bge Base En V1.5	0.89	0.68	0.53	**0.39**	0.39	0.40	0.78	**0.28**	0.88	0.40
AllenAI Scibert Scivocab Uncased	0.18	0.37	0.42	**0.39**	0.39	0.05	0.06	0.27	0.50	0.25
Google BERT Base Uncased	0.27	0.37	0.42	**0.39**	0.39	0.06	0.13	0.26	0.51	0.26
Intfloat E5 Base	0.85	0.65	0.51	**0.39**	0.39	0.38	0.66	0.27	0.79	0.37
Nomic AI Nomic Embed Text V1	**0.90**	0.67	0.57	**0.39**	0.39	0.37	**0.82**	**0.28**	0.88	0.39
Nomic AI Nomic Embed Text V1 Unsupervised	**0.90**	0.71	0.54	**0.39**	0.39	0.37	**0.82**	0.27	0.88	0.42
Sentence-Transformers All MiniLM L6 V2	0.85	0.64	0.55	**0.39**	0.39	0.35	0.68	**0.28**	0.82	0.41
Sentence-Transformers All MPNet Base V2	0.85	0.67	0.51	**0.39**	0.39	0.39	0.69	**0.28**	0.86	**0.45**
Thenlper GTE Base	0.89	0.72	0.53	**0.39**	0.39	**0.42**	0.78	**0.28**	**0.90**	0.43
Medical models										
Abhinand MedEmbed Base	0.89	0.69	0.55	**0.39**	0.39	**0.42**	0.77	**0.28**	0.89	0.44
BioNLP BlueBERT PubMed MIMIC Uncased L-12 768 12	0.09	0.39	0.45	**0.39**	0.39	0.10	0.05	0.26	0.48	0.25
EmilyAlsentzer Bio ClinicalBERT	0.10	0.37	0.43	**0.39**	0.39	0.06	0.06	0.26	0.47	0.23
Kamalkraj BioSimCSE BioLinkBERT Base	0.77	0.56	0.51	**0.39**	0.39	0.20	0.60	0.26	0.82	0.30
Malteos SciNCL	0.78	0.63	0.53	**0.39**	0.39	0.25	0.55	0.27	0.77	0.44
MedicalAI ClinicalBERT	0.13	0.38	0.48	**0.39**	0.39	0.07	0.07	0.26	0.45	0.21
Microsoft BiomedBERT (abstract+full-text)	0.48	0.40	0.45	**0.39**	0.39	0.07	0.24	0.27	0.61	0.27
MedTE Cl15 Step 8000	0.86	**0.81**	**0.61**	**0.39**	**0.45**	**0.42**	0.74	0.27	0.87	**0.45**

Table 4.4: Performance of Embedding Models on Various Sources. All values represent the average metric per task.

Visit the Appendix for further evaluations.

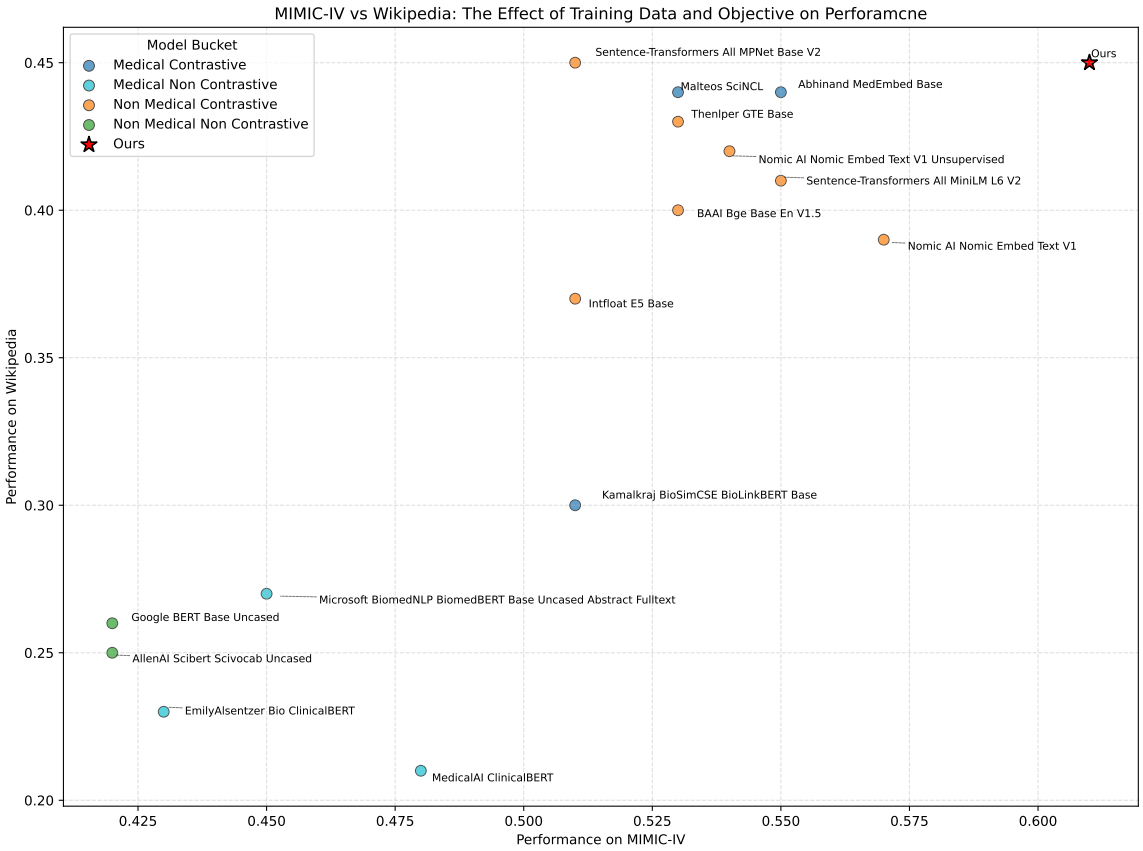


Figure 4.1: Wikipedia vs MIMIC-IV

4.3 Conclusion

In this chapter, we introduced MedTEB, the first large-scale, multi-task benchmark dedicated to medical text embeddings, Across 51 tasks spanning classification, clustering, pair classification, and retrieval, MedTE achieved top-rank performance (0.72 F_1, 0.38 V-measure, 0.74 F_1, 0.45 nDCG@10; overall 0.578), outpacing both general-purpose and existing medical embeddings. Our per-source analysis further demonstrated MedTE's robustness on medical data, while maintaining strong results on general-domain text (Wikipedia). These findings underscore two key takeaways: (1)

domain-adaptive contrastive learning is essential for capturing the fine-grained semantics of medical language, and (2) a unified, comprehensive benchmark like MedTEB is critical for reliably evaluating embedding quality across real-world healthcare applications. By open-sourcing MedTEB, we lay a foundation for future medical NLP research.

Limitations While MedTEB demonstrates strong performance across various chemical text embedding tasks, several limitations remain. First, the benchmark does not currently incorporate varying levels of task difficulty, which constrains its ability to evaluate model robustness under progressively challenging scenarios. Second, the dataset's size and domain diversity are limited, potentially restricting its representativeness of the full spectrum of real-world chemical text. Additionally, in LLM-based tasks, particularly pair classification, the employed prompting strategy may inadvertently influence the writing style of generated text, introducing potential biases in evaluation outcomes, and the quality of the generated questions was not systematically verified. Moreover, no ablation study was conducted to systematically analyze how individual factors, such as prompt design, dataset choices, or evaluation setup, contributed to the overall results, and the evaluation relied solely on GPT-4o, without examining how other models or alternative approaches to substituting LLMs might impact the findings. Finally, systematic comparisons with classical pair-finding methods have not yet been conducted, leaving a gap in establishing comprehensive performance baselines.

Future Work Future work could involve expanding the dataset by incorporating more data from diverse and complementary sources, enabling the creation of more

complex tasks and a wider range of difficulty levels for a deeper assessment of model capabilities. The benchmark could also explore more fine-grained difficulty categorization to better understand performance across task complexity gradients. In addition, improvements to LLM prompting strategies could be made in the task generation process to reduce stylistic biases and improve consistency. Finally, Ablation studies on different components of the task generation pipeline, especially the parts that involve LLMs, could further clarify the impact of specific design choices.

4.4 Attribution

Citation M. Khodadad, A. Shiraee Kasmaee, M. Astaraki, and H. Mahyar, "Towards Domain Specification of Embedding Models in Medicine," *arXiv*, Jul. 2025, arXiv:2507.19407. [Online]. Available: https://arxiv.org/abs/2507.19407 [80]

Contributions This project was the result of collaborative work. Mohammad Khodadad led the ideation, benchmark preparation, and writing. Ali Shiraee contributed to the ideation and provided consultation throughout the project, while Mahdi Astaraki assisted with prompt tuning, benchmarking, and writing. Dr. Hamidreza Mahyar supervised the project and provided overall guidance.

Resources All experiments were conducted on Compute Canada infrastructure, utilizing 64 CPU cores and 150 GB RAM. The software environment included `Python`, `git`, `openai`, `MTEB`, and `Sentence-Transformers` for optimized evaluation, with package management handled via `pip`. Code is available at `https://zenodo.org/records/16882534` and `https://github.com/MohammadKhodadad/MedTEB-dev`

Chapter 5

ChemTEB

5.1 Introduction

The application of deep learning and transformer-based architectures has led to substantial advances in natural language understanding, yet the unique linguistic and semantic challenges of chemical literature remain under-addressed. Early text representation methods such as Word2Vec [110] and GloVe [135] capture word co-occurrence statistics, but they lack the contextual sensitivity required for domain-specific terminology. The advent of self-attention in Transformers [182] and models like BERT [43] and RoBERTa [103] improved contextual embeddings, while domain-tuned variants such as SciBERT [15] began to close the gap for scientific text. However, even recent contrastive-learning approaches (e.g. E5 [188], Nomic Embed [125]) and multi-granular models (M^3-Embed [119]) are primarily evaluated on general or broad-science benchmarks like MTEB [116].

Chemical NLP poses distinct challenges: SMILES and IUPAC strings follow rigid syntactic rules, while textual descriptions in patents, literature, and safety data sheets

exhibit dense domain-specific jargon and abbreviations. Embeddings must therefore capture both structural nuances of chemical identifiers and the semantic context of narrative descriptions. Moreover, applications such as reaction prediction, patent retrieval, and cheminformatics-driven literature mining demand high precision and recall of chemically meaningful concepts.

To address these needs, we introduce the Chemical Text Embedding Benchmark (ChemTEB), a comprehensive evaluation suite designed explicitly for the chemical sciences. ChemTEB comprises tasks spanning chemical text classification, similarity assessments between natural language and SMILES pairs, clustering of reaction descriptions, and retrieval of protocol or spectral data. By evaluating 34 open-source and proprietary models, including both generic and chemistry-tuned embeddings, ChemTEB reveals the strengths and limitations of current methodologies in processing chemical information. Our benchmark, accompanied by open-source code and data, provides a standardized, domain-specific framework to guide the development of more accurate, efficient NLP models for chemistry applications.

5.2 ChemTEB

The Chemical Text Embedding Benchmark (ChemTEB) evaluates embedding models on a suite of chemistry-focused tasks, leveraging heterogeneous datasets that capture both structured and unstructured chemical knowledge. We draw from five primary sources: the PubChem compound database for molecular property and similarity tasks [83], English Wikipedia for standardized chemical concept descriptions, the BEIR retrieval benchmark for text-based search evaluations [175], CoconutDB for year-extracted reaction and synthesis data [169], and industry Safety Data Sheets

(SDS) for domain-specific terminology and hazard classification [136]. Each task and its associated dataset has been curated or validated in collaboration with professional chemists to ensure that ChemTEB reflects real-world requirements in chemical information retrieval, classification, and similarity assessment.

5.2.1 Tasks

ChemTEB comprises five task categories, each designed to probe distinct aspects of chemical text understanding:

- **Classification**: Assigning documents (e.g. SDS entries or Wikipedia abstracts) to predefined chemical categories such as hazard classes or compound families.
- **Pair Classification**: Determining semantic equivalence between natural-language descriptions and SMILES representations, or between reaction step descriptions.
- **Clustering**: Grouping related chemical documents or compound descriptions into coherent clusters, evaluated using metrics like V-measure.
- **Retrieval**: Ranking relevant chemical literature or protocol documents given a query (e.g. retrieving synthesis procedures from CoconutDB or PubChem entries based on text queries).
- **Bitext Mining**: Aligning parallel corpora of SMILES strings and their corresponding textual descriptions to measure alignment quality.

For each benchmark, we describe the source dataset, preprocessing steps, input–output format, and evaluation metric. Table 5.1 summarizes the dataset sizes,

label schemas, and key statistics for all tasks in ChemTEB, providing a concise reference for model comparison and further extension.

Table 5.1: Datasets summary. This table provides an overview of the datasets used across different tasks, including the dataset names from Hugging Face, the original data sources, and the distribution of sample sizes. The distribution is represented through key statistical measures: 5th percentile, median, and 95th percentile of the number of tokens

Task		HuggingFace Name	Data Source	#Samples	Sequence Lengths (tokens [1])		
					5th Percentile	Median	95th Percentile
Classification	1	WikipediaEasy10Classification	Wikipedia	2105	42	178	612.4
	2	WikipediaEasy5Classification	Wikipedia	1164	43	171.5	547.85
	3	WikipediaMedium5Classification	Wikipedia	617	39	137	563.6
	4	WikipediaMedium2CrystallographyVsChromatographyTitrationpHClassification	Wikipedia	1451	41.5	175	658.5
	5	WikipediaMedium2BioluminescenceVsNeurochemistryClassification	Wikipedia	486	42	158	574.25
	6	WikipediaEZ2Classification	Wikipedia	58921	41	164	590
	7	WikipediaHard2BioluminescenceVsLuminescenceClassification	Wikipedia	410	41	148.5	579.3
	8	WikipediaEasy2GeneExpressionVsMetallurgyClassification	Wikipedia	5741	42	175	630
	9	WikipediaEasy2GreenhouseVsEnantiopureClassification	Wikipedia	1136	34	139.5	513
	10	WikipediaEZ10Classification	Wikipedia	43146	41	165	582
	11	WikipediaHard2SaltsVsSemiconductorMaterialsClassification	Wikipedia	491	38.5	141	447.5
	12	WikipediaEasy2SolidStateVsColloidalClassification	Wikipedia	2216	42	151	532
	13	WikipediaMedium2ComputationalVsSpectroscopistsClassification	Wikipedia	1101	38	155	639
	14	WikipediaHard2IsotopesVsFissionProductsNuclearFissionClassification	Wikipedia	417	43.8	209	706.4
	15	WikipediaEasy2SpecialClassification	Wikipedia	1312	35.55	133	465
	16	SDSGlovesClassification	Safety Data Sheets	8000	498	1071	1871
	17	SDSEyeProtectionClassification	Safety Data Sheets	8000	492	1060	1876
BitextMining	18	CoconutSMILES2FormulaBM	CoconutDB	8000	6	11	150
	19	PubChemSMILESISoTitleBM	PubChem	14140	4	22	93
	20	PubChemSMILESISoDescBM	PubChem	14140	12	45	134
	21	PubChemSMILESCanonTitleBM	PubChem	30914	3	12	43
	22	PubChemSMILESCanonDescBM	PubChem	30914	8	24	109
Retrieval	23	ChemHotpotQARetrieval	HotpotQA	10275	19	71	183
	24	ChemNQRetrieval	Natural Questions	22960	13	81	231
Clustering	25	WikipediaMedium5Clustering	Wikipedia	617	39	137	563.6
	26	WikipediaEasy10Clustering	Wikipedia	2105	42	178	612.4
PairClassification	27	WikipediaAIParagraphsParaphrasePC	Wikipedia	5408	28	104	354
	28	CoconutSMILES2FormulaPC	CoconutDB	8000	6	11	108
	29	PubChemAISentenceParaphrasePC	PubChem	4096	9	20	59
	30	PubChemSMILESCanonTitlePC	PubChem	4096	4	16	30
	31	PubChemSynonymPC	PubChem	4096	3	8	38
	32	PubChemSMILESCanonDescPC	PubChem	4096	12	23	105
	33	PubChemSMILESIsoDescPC	PubChem	4096	12	48	125
	34	PubChemSMILESIsoTitlePC	PubChem	4096	4	35	70
	35	PubChemWikiParagraphsPC	PubChem	4096	8	66	235

Classification. Each dataset comprises a text field and corresponding labels. We fine-tune each embedding model on the training split and train a logistic regression classifier on the resulting embeddings. Performance is evaluated on the test split using macro-averaged F_1 [108]. Classification datasets are drawn from: (i) chemistry-related English Wikipedia articles categorized into subfields, and (ii) Safety Data Sheets (SDS) providing detailed chemical hazard and property information [136].

Clustering. We group related text segments into coherent clusters using Mini-Batch

k-means (batch size 32) on sentence embeddings. Clustering data are constructed from Wikipedia article sections, with cluster quality measured by V-measure [154].

Pair Classification. This binary task determines if two texts refer to the same chemical entity or match compound descriptions to their SMILES representations. We embed each pair, compute four similarity metrics (cosine, Euclidean, Manhattan, dot product), select the optimal threshold on the training set, and report the maximum F_1 across metrics. Datasets originate from PubChem [83] and COCONUT [169].

Bitext Mining. We align pairs of semantically equivalent texts, SMILES strings and their natural-language descriptions, by ranking corpus embeddings against query embeddings using cosine similarity. Data sources include PubChem [83] and COCONUT [169]. Performance is measured by F_1 over correctly retrieved pairs.

Retrieval. Each retrieval dataset consists of queries and a document corpus with relevance labels. We embed all texts and rank documents by cosine similarity. A chemistry-focused subset of Natural Questions [86] and HotpotQA [207] is used, with nDCG@10 as the primary metric.

5.2.2 Embedding Models

We evaluate 34 embedding models (27 open-source, 7 proprietary) spanning general and chemistry-specific architectures. Refer to the appendix for the summarization of model sizes, training objectives, and domain specializations. (Detailed specifications are shown in the Appendix.)

5.2.3 Ranking Process for Model Performance

Models are ranked within each task category by computing the arithmetic mean of performance metrics across all datasets. To aggregate across categories, we apply Reciprocal Rank Fusion (RRF) [39]:

$$\text{RRF\_score}(m) \;=\; \sum_{d \in \text{Datasets}} \frac{1}{k + r_d(m)},$$

where $r_d(m)$ is the rank of model m on dataset d, and $k = 10$ is a constant dampening factor. The final RRF score reflects a model's overall ranking across all ChemTEB tasks, with higher scores indicating more consistent high performance.

5.3 Results

5.3.1 Model Performance

Table 5.3.1 presents each model's average score per task category and its overall ranking ($\text{RRF}_{\text{score}}$). From a **model** perspective, no single architecture dominates every task; however, proprietary embeddings generally outperform open-source counterparts. Notably, *OpenAI-Text Embedding 3-Large* ranks first in three of five task categories [26], while among open-source solutions, *Nomic Embed Text V1.5* achieves the best overall $\text{RRF}_{\text{score}}$ [125].

	Classification (Macro F1)	Bitext Mining (F1)	Retrieval (nDCG@10)	Clustering (V-measure)	Pair Classification (Max F1)	Final Score (RRF)
BERT	0.72±0.04	0.0±0.0	0.28±0.02	0.2±0.03	0.41±0.05	0.122
SciBERT	0.71±0.04	0.0002±0.0	0.2±0.03	0.18±0.02	0.43±0.05	0.122
MatSciBERT	0.7±0.04	0.0003±0.0001	0.11±0.02	0.21±0.03	0.41±0.05	0.122
Chemical BERT	0.68±0.04	0.0003±0.0	0.17±0.01	0.13±0.02	0.42±0.05	0.120
Nomic BERT	0.67±0.04	0.0001±0.0	0.05±0.0	0.22±0.03	0.38±0.04	0.118
Nomic Embedding v1	0.77±0.04	0.0023±0.0002	0.72±0.02	0.46±0.03	**0.55±0.06**	0.285
Nomic Embedding v1.5	0.78±0.04	0.0026±0.0002	0.75±0.02	0.5±0.04	**0.55±0.06**	0.339
SBERT - all Mini LM L6.v2	0.78±0.03	0.0015±0.0002	0.61±0.01	0.36±0.02	0.54±0.06	0.232
SBERT - all Mini LM L12.v2	0.77±0.04	0.0013±0.0001	0.58±0.0	0.34±0.01	0.54±0.06	0.201
SBERT - all MPNET-base.v2	0.78±0.04	0.001±0.0001	0.56±0.0	0.5±0.03	0.54±0.06	0.239
SBERT - multi-qa-mpnet-base.v1	0.74±0.04	0.0009±0.0001	0.56±0.01	0.42±0.04	0.54±0.06	0.185
E5 - small	0.75±0.03	0.0015±0.0001	0.69±0.02	0.12±0.02	0.48±0.05	0.166
E5 - base	0.76±0.04	0.0019±0.0001	0.68±0.01	0.34±0.05	0.49±0.05	0.192
E5 - large	0.77±0.04	0.0029±0.0002	0.7±0.01	0.51±0.04	0.5±0.05	0.290
E5 - small v2	0.76±0.03	0.0012±0.0001	0.69±0.01	0.19±0.03	0.46±0.05	0.165
E5 - base v2	0.76±0.04	0.0016±0.0001	0.68±0.01	0.38±0.05	0.47±0.05	0.178
E5 - large v2	0.76±0.04	0.0022±0.0002	0.73±0.01	0.33±0.05	0.48±0.05	0.214
E5 - Multilingual small	0.74±0.04	0.0018±0.0001	**0.76±0.01**	0.17±0.01	0.47±0.05	0.207
E5 - Multilingual base	0.75±0.04	0.0022±0.0001	0.68±0.0	0.48±0.03	0.47±0.05	0.196
E5 - Multilingual large	0.74±0.04	0.0026±0.0002	0.67±0.0	0.3±0.05	0.48±0.05	0.187
BGE - small en	0.78±0.04	0.0012±0.0001	0.52±0.04	0.27±0.03	0.48±0.05	0.160
BGE - base en	0.77±0.04	0.0019±0.0001	0.59±0.03	0.44±0.05	0.48±0.05	0.186
BGE - large en	0.78±0.04	0.0016±0.0001	0.44±0.06	0.45±0.05	0.49±0.05	0.191
BGE - small en v1.5	0.78±0.03	0.0013±0.0001	0.63±0.03	0.25±0.04	0.48±0.05	0.180
BGE - base en v1.5	0.77±0.04	0.0018±0.0001	0.69±0.02	0.47±0.05	0.49±0.05	0.219
BGE - large en v1.5	0.78±0.04	0.0019±0.0001	0.67±0.02	0.39±0.06	0.5±0.05	0.224
BGE - Multilingual - M3	0.76±0.03	0.0012±0.0002	0.68±0.02	0.45±0.05	0.47±0.06	0.176
OpenAI - Text embedding 3 - small	0.78±0.04	0.0027±0.0003	0.65±0.01	0.49±0.05	0.5±0.05	0.273
OpenAI - Text embedding 3 - large	0.8±0.04	**0.0062±0.0006**	0.71±0.01	**0.6±0.03**	0.53±0.05	**0.384**
OpenAI - Text embedding - Ada - 02	0.78±0.04	0.0035±0.0002	0.66±0.02	0.52±0.04	0.49±0.05	0.279
Amazon - Titan Text Embedding v2	0.77±0.03	0.0024±0.0002	0.62±0.0	0.49±0.04	0.49±0.05	0.224
Amazon - Titan Embedding G1 Text	**0.81±0.03**	0.0032±0.0003	0.6±0.02	0.45±0.06	0.49±0.05	0.285
Cohere - Embed English V3	**0.81±0.03**	0.0012±0.0	0.49±0.04	0.55±0.02	0.53±0.06	0.278
Cohere - Embed Multilingual V3	0.8±0.03	0.0024±0.0001	0.49±0.04	0.53±0.03	0.53±0.06	0.281

Considering **task** difficulty, classification exhibits the highest scores, reflecting the

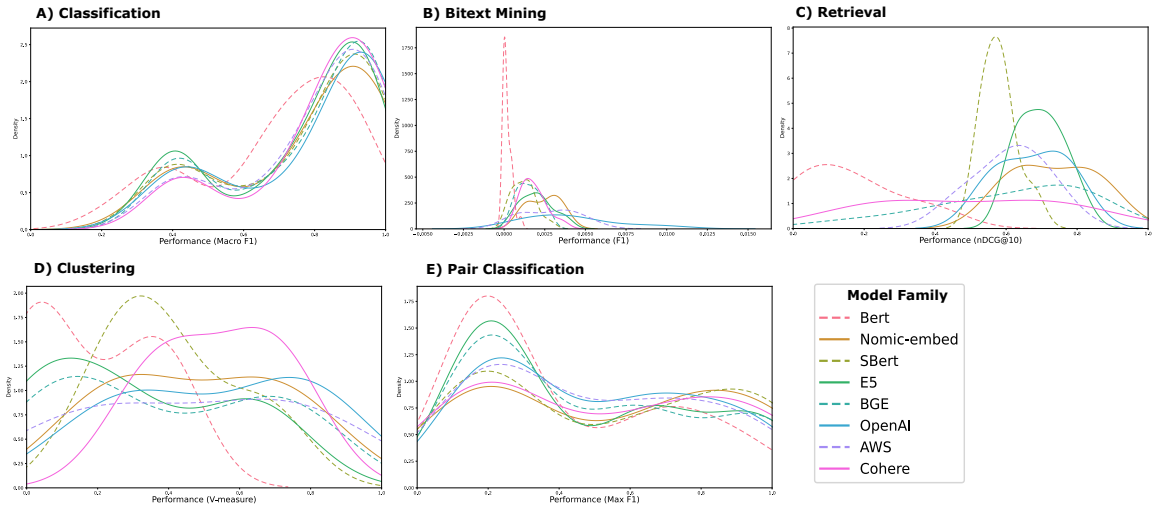


Figure 5.1: Distribution plots for five categories of tasks. The KDE plots show the probability density functions, where the x-axis represents the range of predicted values (performance distribution over tasks of each category and models of each family) and the y-axis represents the estimated density. Each colored line corresponds to a unique model family, enabling a clear visual comparison of their value distributions.

relative simplicity of assigning discrete labels [108]. In contrast, bitext mining, aligning SMILES strings with their textual descriptions, yields the lowest performance, as general-purpose models lack training on chemical notation such as SMILES. Retrieval, clustering, and pair classification occupy intermediate positions in descending order of difficulty and average model performance.

To analyze the influence of architectural family on performance, we group models into eight clusters: BERT-family [43], Nomic-family [125], SBERT-family [149], E5-family [188], BGE-family [201], OpenAI-family [26], Amazon-family, and Cohere-family. Figure 5.1 visualizes each family's score distribution across tasks using kernel density estimation [133], highlighting how contrastively trained groups (e.g. Nomic, E5) often outperform purely MLM-pretrained families.

5.3.2 Model Efficiency

Embedding models vary widely in parameter count, embedding dimension, and inference speed. Figure 5.2 plots each model's pair-classification throughput (queries/sec), model size (MB), embedding dimension, and overall $\text{RRF}_{\text{score}}$. A clear trend emerges: larger models tend to be slower but achieve higher $\text{RRF}_{\text{score}}$. For instance, *OpenAI-Text Embedding 3-Large* secures the top $\text{RRF}_{\text{score}}$ yet exhibits the lowest throughput. Conversely, *SBERT-All-MiniLM-L6-v2* combines minimal model size and embedding dimension with the highest speed, albeit at reduced performance [149]. *Nomic Embed Text V1.5* strikes a favorable balance between speed and accuracy, making it a strong open-source candidate for time-sensitive chemical NLP tasks [125].

These efficiency results offer practitioners a pragmatic guide: choose larger, contrastively trained models when peak accuracy is required, but prefer compact models for real-time or resource-constrained chemical information processing.

5.3.3 Domain Adaptation

To date, only a few embedding models have been explicitly adapted to chemistry. MatSciBERT [51] and ChemicalBERT[2] extend the BERT architecture with domain-specific pretraining on chemical corpora, while SciBERT [15], though trained on broad scientific text, also shows some chemical capacity. In our bitext mining task, which requires partial SMILES understanding, these BERT-family adaptations outperform vanilla BERT-Base. However, their gains do not generalize: outside bitext mining, they fail to deliver consistent improvements, and collectively they occupy the lowest

[2] `https://huggingface.co/recobo/chemical-bert-uncased`

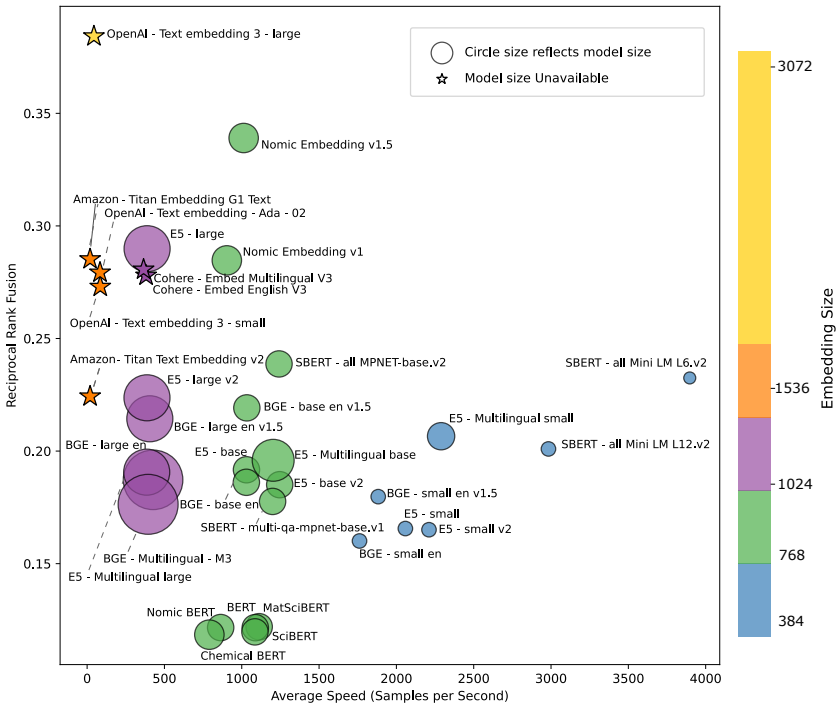


Figure 5.2: Summary of evaluated models in terms of efficiency. All evaluated models are depicted in the form of (i) circles (with circle size being proportional to the number of parameters) for open-source models, and (ii) stars for proprietary models. The color of the depicted models reflects their embedding dimension. The x-axis denotes the averaged inference speed (embedded samples/sec) calculated over seven pair classification tasks (tasks 29 - 35 in table 5.1) conducted on a V100 GPU machine.

$\text{RRF}_{\text{score}}$ positions (see Supplementary). This suggests that simple domain adaptation of an MLM-only architecture is insufficient for the varied demands of ChemTEB. Instead, our results imply that contrastive objectives and architectural enhancements developed after BERT, seen in newer families like E5, BGE, and Nomic, drive greater semantic discrimination in chemistry. We therefore encourage future work to prioritize modern, contrastively trained designs when specializing embeddings for chemical applications.

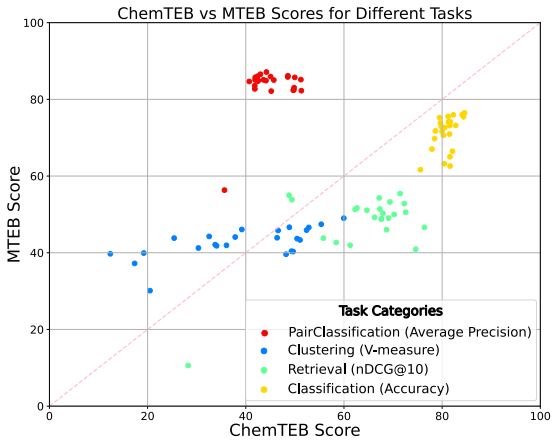


Figure 5.3: Performance comparison between ChemTEB and MTEB benchmarks across task categories. Each point corresponds to a model evaluated on both suites, highlighting domain-specific difficulty and the impact of specialized pretraining.

5.4 Conclusion

In this chapter, we presented ChemTEB, the first open-source benchmark tailored to chemical text embeddings, and analyzed 34 embedding models across classification, pair classification, clustering, retrieval, and bitext mining. Our extensive evaluations reveal that proprietary, contrastively trained models generally outperform open-source alternatives, and that simple MLM-based domain adaptation (e.g. MatSciBERT, ChemicalBERT) yields only limited gains. By providing a standardized suite of 36 datasets validated by chemists, ChemTEB enables rigorous comparison and drives the development of more precise, efficient NLP tools for chemistry. We make all code and data publicly available, and we hope ChemTEB will serve as a foundation for future work in chemical NLP, including multimodal integration, causal reasoning, and retrieval-grounded generation in chemical discovery.

Limitations While ChemTEB covers a wide range of evaluation tasks, the current benchmark is skewed toward classification-oriented datasets, with relatively fewer resources dedicated to retrieval. Additionally, a substantial portion of the benchmark's text data is sourced from Wikipedia, which, while well-structured, may not fully reflect the complexity, diversity, and style of real-world chemical literature, patents, or experimental records. This concentration in both task type and data source may limit the generalizability of performance rankings, particularly for models intended for retrieval-based or generative chemical NLP applications.

Future Work Future iterations of ChemTEB could expand coverage by incorporating datasets from diverse, domain-rich sources such as scientific journals, patents, laboratory protocols, and chemical safety reports. Increasing the proportion of retrieval and synthesis-related tasks would enable a more balanced assessment of embedding models across real-world chemical workflows. In parallel, developing a chemistry-specialized embedding model trained via contrastive learning on large, heterogeneous chemical corpora, with chemically informed tokenization, could narrow the performance gap with proprietary systems. Further gains may come from integrating these embeddings into multimodal pipelines that combine text with molecular graphs, reaction schemes, or spectroscopic data, supporting advanced tasks in discovery and analysis.

5.5 Attribution

Citation A. Shiraee Kasmaee, M. Khodadad, M. A. Saloot, N. Sherck, S. Dokas, S. Samiee, and H. Mahyar, "ChemTEB: Chemical Text Embedding Benchmark, an

Overview of Embedding Models Performance Efficiency on a Specific Domain," arXiv preprint arXiv:2412.00532, 2024. [Online]. Available: https://arxiv.org/abs/2412.00532 [78]

Contributions This project was carried out through collaborative efforts. Mohammad Khodadad contributed to the ideation, task preparation, visualizations, and writing of the paper. Ali Shiraee contributed to ideation, task preparation, benchmarking, and writing. Mohammad Arshi Saloot assisted with task preparation and writing. Nicholas Sherck and Stephen Dokas supported the chemical verification of tasks and benchmarks. Soheila Samiee and Dr. Hamidreza Mahyar supervised the project and provided overall guidance.

Resources All experiments were conducted on Compute Canada infrastructure, utilizing 64 CPU cores and 150 GB RAM. The software environment included `Python`, `git`, `openai`, `MTEB`, and `Sentence-Transformers` for optimized evaluation, with package management handled via `pip`. Code is available at `https://zenodo.org/records/16896164` and `https://github.com/basf/chemteb`

Chapter 6

Evaluating Multi-Hop Reasoning in Large Language Models: A Chemistry-Centric Case Study

6.1 Introduction

Large language models (LLMs) excel at many language tasks but continue to struggle with compositional, multi-step reasoning, particularly in specialized domains like chemistry where inference must traverse complex relational chains. Chain-of-thought prompting and structural enhancements (e.g. CoT [192, 190, 211], RAG [95], neuro-symbolic methods [157]) have improved reasoning in general domains, yet chemical reasoning benchmarks remain scarce and limited in scope [194, 70].

To fill this gap, we introduce GraphRAG, an automated pipeline that constructs domain-specific knowledge graphs from recent chemical literature and assesses LLM performance on challenging multi-hop question-answering. We first apply named

entity recognition (NER) and LLM-based Extraction to extract chemical entities and their relations from unstructured text [87]. These entities are linked to external resources (e.g. PubChem, ChEBI) to form a richly connected knowledge graph. Next, GraphRAG generates multi-hop questions by sampling paths of varying lengths through the graph, ensuring each query requires compositional inference across multiple edges.

We evaluate both context-augmented (with retrieved graph facts) and non-augmented settings, measuring accuracy and reasoning fidelity. Our experiments demonstrate that even perfect retrieval of relevant facts does not guarantee correct multi-step reasoning, highlighting intrinsic limitations of current LLM architectures in domain-specific compositional tasks. By providing a fully automated, expert-validated benchmark and data generation pipeline, GraphRAG offers a scalable framework for probing and advancing chemical reasoning capabilities in state-of-the-art language models.

6.2 Methodology

Our approach to evaluating compositional reasoning in chemistry comprises three key stages: (i) constructing a domain-specific knowledge graph, (ii) generating multi-hop question–answer pairs from that graph, and (iii) assessing state-of-the-art LLMs on the resulting QA tasks. The first two components are detailed below; the third is presented in Experiments. Visit the Appendix for a detailed explanation of these steps, including the used LLM prompts.

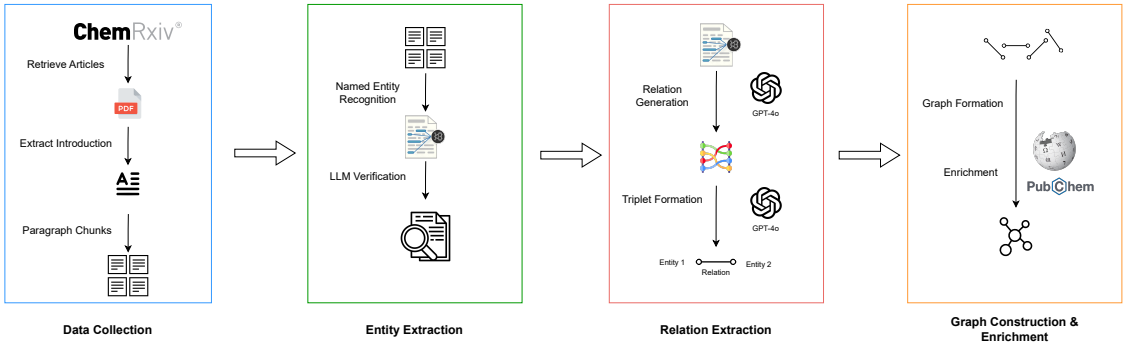


Figure 6.1: An Overview of the knowledge graph generation pipeline.

6.2.1 Knowledge Graph Generation

As shown in figure 6.1, we automatically build a chemical knowledge graph from recent ChemRxiv preprints. Using the ChemRxiv API, we retrieve all articles with redistribution-compatible licenses, then extract each article's introduction (up to 500 words) via regex-based cleaning. Introductions are split into contiguous chunks of up to 128 words, ensuring paragraph integrity.

Each text chunk is processed with a PubMedBERT-based NER model [155, 49] fine-tuned on chemical entity recognition to identify mentions of reagents, products, catalysts, and other domain entities. Extracted entities are then reviewed and refined by OpenAI's `gpt-4o` to ensure chemical validity and specificity. Relations between entities (e.g. "catalyzes", "reacts_with") are likewise extracted using `gpt-4o`, yielding subject–predicate–object triplets. To enrich each node, we augment with metadata and descriptive annotations sourced from Wikipedia and PubChem [82]. The final graph comprises nodes representing chemically verified entities (with attached descriptions and identifiers) and edges encoding the extracted relations. Figure 6.1 illustrates the pipeline for graph construction.

6.2.2 Multi-hop Question-Answer Generation

As shown in figure 6.2, To probe compositional reasoning, we sample multi-edge paths via a randomized breadth-first search over the knowledge graph that enforces distinct source documents per edge. A path of length K thus spans $K+1$ entities drawn from K unique ChemRxiv chunks.

For each edge (e_i, r_i, e_{i+1}), we first generate a one-hop question: "What entity e_i satisfies the relation r_i with e_{i+1}?" When necessary, `o3-mini` enriches questions with contextual metadata to ensure unambiguous prompts. We then chain these sub-questions into a single multi-hop query by reversing the hop order: starting from the final relation and appending each previous step, guaranteeing that the ultimate answer corresponds to e_1. `o3-mini` also validates the logical coherence of the assembled question and refines phrasing for clarity.

To prevent spurious or unsolvable items, any question that all target models fail to answer correctly is discarded. The resulting dataset thus consists of rigorously validated, context-rich multi-hop questions that require stepwise integration of disjoint graph facts. Figure 6.2 depicts the end-to-end QA generation workflow.

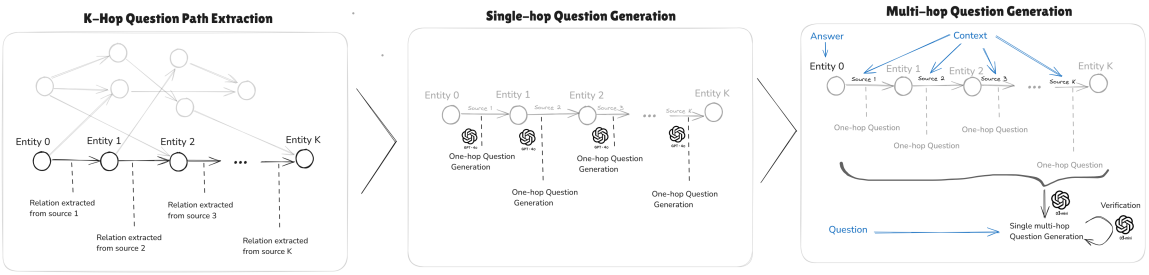


Figure 6.2: Overview of the QA generation Pipeline.

Model	Context	Correctness Rate (%)	Avg Duration (s)	Avg Input Tokens	Avg Output Tokens	Total Input Tokens (K)	Total Output Tokens (K)
Anthropic Claude Sonnet 3.5 V2	✗	40.06	1.54	567	29	550.93	28.69
Anthropic Claude Sonnet 3.5 V2	✓	72.50	1.68	2210	30	2146.11	29.18
Anthropic Claude Sonnet 3.7	✗	44.80	1.61	567	30	550.93	29.35
Anthropic Claude Sonnet 3.7	✓	80.02	1.84	2210	30	2146.11	29.49
Anthropic Claude Sonnet 3.7 (Thinking)	✗	45.73	39.01	583	1777	566.09	1725.79
Anthropic Claude Sonnet 3.7 (Thinking)	✓	84.35	15.35	2228	715	2163.59	694.78
OpenAI GPT-4o-mini	✗	32.34	0.63	204	9	198.60	9.63
OpenAI GPT-4o-mini	✓	62.82	0.71	1628	10	1581.57	10.01
OpenAI GPT-4o	✗	40.27	0.63	204	9	198.60	9.53
OpenAI GPT-4o	✓	68.80	0.71	1628	10	1581.57	9.95
OpenAI o1-mini	✗	41.09	7.78	160	1047	155.88	1017.55
OpenAI o1-mini	✓	71.99	5.68	1609	718	1562.70	697.41
OpenAI o3-mini	✗	47.58	10.84	210	1187	204.43	1153.12
OpenAI o3-mini	✓	80.33	6.12	1634	558	1587.40	542.46
Mistral Large	✗	35.53	0.41	177	13	172.45	13.40
Mistral Large	✓	73.94	0.57	1913	14	1857.70	14.22
Llama 3.3 70B Instruct	✗	32.13	0.33	330	10	320.47	10.56
Llama 3.3 70B Instruct	✓	65.19	0.40	1781	11	1729.91	10.75
Google Gemma 3 27B	✗	32.03	0.89	163	12	158.95	11.94
Google Gemma 3 27B	✓	69.72	1.00	1587	12	1541.55	12.57
DeepSeek R1	✗	44.39	21.06	159	1466	154.40	1423.73
DeepSeek R1	✓	81.98	8.61	1551	573	1506.14	556.55
Qwen QwQ 32B	✗	35.74	68.29	168	2167	163.51	2104.86
Qwen QwQ 32B	✓	79.81	25.18	1665	757	1617.45	735.86
DeepSeek R1 Distill Qwen 32B	✗	34.19	32.04	159	1074	154.70	1043.56
DeepSeek R1 Distill Qwen 32B	✓	79.09	12.11	1633	400	1586.25	389.31

Table 6.1: Summary of tested models' performance in terms of several evaluation metrics for both Contextual and Non-Contextual Setups

6.3 Experiments and Results

6.3.1 Models Performance

In our experiments, we assessed the domain-specific multi-hop question-answering performance of 13 state-of-the-art large language models, encompassing both reasoning-oriented and general-purpose variants. Models optimized for test-time compute are labeled **reasoning models**. We tested each model with and without retrieval-provided context to simulate ideal RAG and closed-book settings. All OpenAI models (`gpt-4o`, `gpt-4o-mini`, `o1-mini`, `o3-mini`) were accessed via the OpenAI API; Anthropic and Mistral family models via Amazon Bedrock; and Google Gemma, Qwen QwQ, and distilled DeepSeek via OpenRouter. Each model was prompted for JSON output to enable automatic correctness checking. We measured the *Correctness Rate* (%) over 971 questions spanning 1–4 hops (avg. 245 per hop). Table 6.1 summarizes overall performance.

Figure 6.3 plots correctness rate, cost, and latency under both context-provided

and context-not-provided conditions. In the RAG setting, `Llama 3.3 70B Instruct` and `GPT-4o` achieve minimal cost and latency but the lowest accuracy, offering a cost-efficient yet less reliable option. Conversely, `Claude Sonnet 3.7 (extended)` attains the highest correctness at significantly greater cost and latency. Qwen QWQ 32B and DeepSeek R1 Distill QWQ 32B strike favorable cost-accuracy trade-offs in the RAG setup, though with above-average latency. In the closed-book setting, open-source reasoning models (`R1 Distill Qwen`, `QWQ-32B`, `R1`) underperform relative to closed-source variants, suggesting broader pre-training benefits. Extended thinking for Claude 3.7 offers no advantage without context, instead increasing token usage and cost. Detailed metrics appear in Table 6.1.

6.3.2 Comparison with HotpotQA and ChemlitQA

To contextualize our results, we extracted a chemistry-specific subset of HotpotQA [207] by filtering questions whose Wikipedia titles fall under the Chemistry category and its subcategories. We evaluated models on this subset using only supporting documents as context. Figure 6.4 compares average correctness rates with and without context. With context, model performance is similar across benchmarks, though our ChemRxiv-derived dataset yields slightly lower averages and reduced variance. Without context, models struggle more on our ChemMultiHop dataset than on HotpotQA, likely because HotpotQA's Wikipedia source overlaps with model pre-training, whereas our dataset draws from recent ChemRxiv papers augmented with PubChem and Wikipedia metadata.

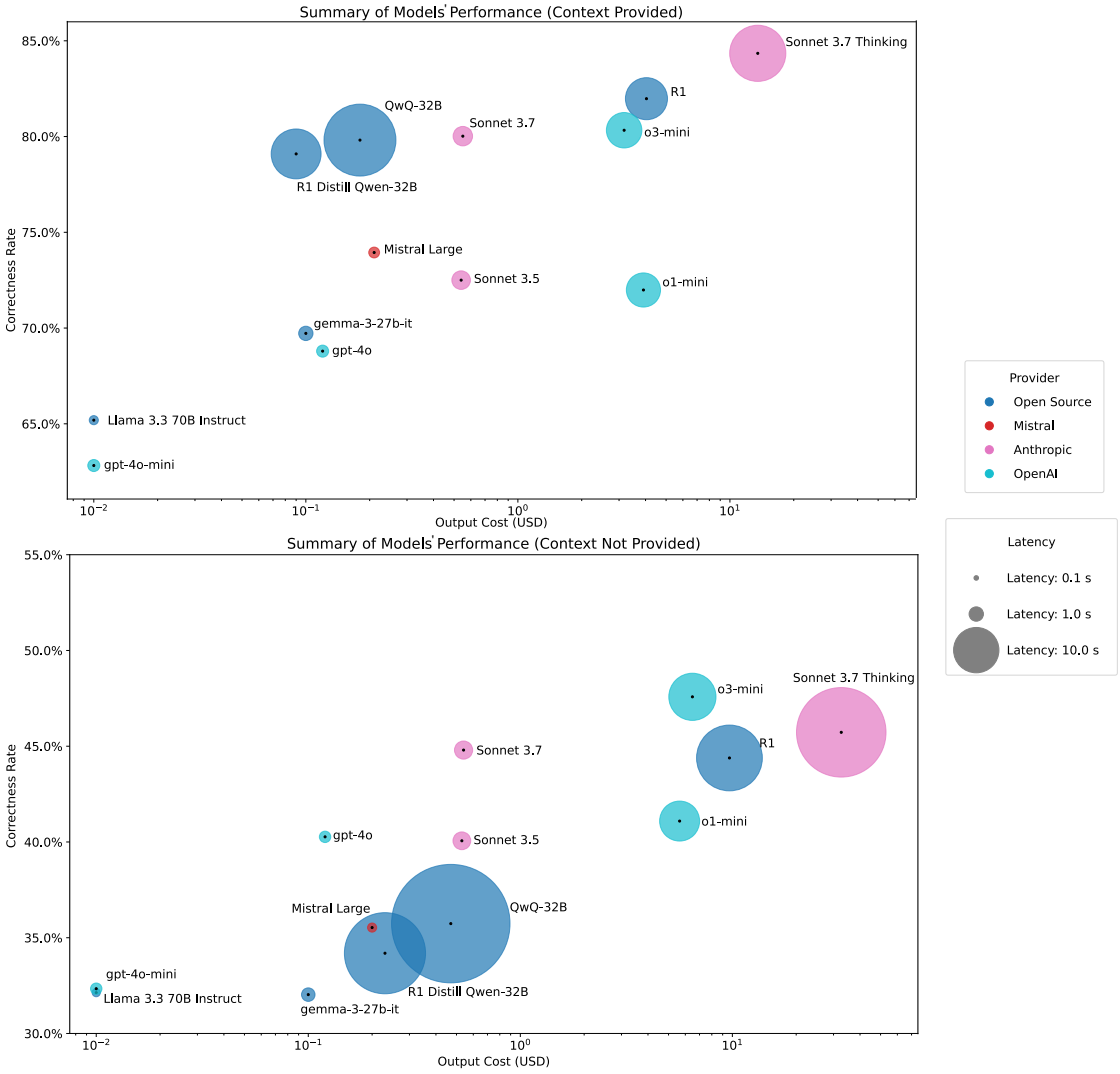


Figure 6.3: Performance of selected models is shown in terms of correctness rate, cost, and latency. The cost axis uses a logarithmic scale to highlight differences. The y-axis indicates the percentage of questions each model answers correctly, and the size of each dot reflects the model's average latency when responding. The top panel shows results for setups where context is provided, and the bottom panel shows results for setups without context. The horizontal axis range is the same in both panels, but the vertical axis ranges differ.

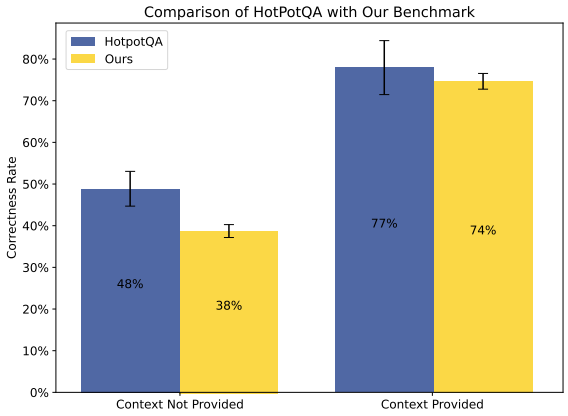


Figure 6.4: Comparison of LLM performance on the chemical subset of HotPotQA versus the curated QA dataset from this study. Error bars indicate the standard error of the mean (S.E.M) across the evaluated models.

6.4 Analysis and Ablation

This section provides detailed ablation studies and benchmark analyses. We begin by investigating how context availability and test-time reasoning influence model performance and efficiency. We then analyze how the number of reasoning hops, serving as a measure of question difficulty, affects both model accuracy and the token count required to produce an answer.

6.4.1 Dataset- and Graph-level Statistics

Table 6.2 provides a concise overview of both our dataset-level and graph-level statistics. In Table 6.2a, we summarize key properties of the 971 multi-hop questions, including average question and answer lengths (in characters and tokens), the mean number of hops per question, total and pooled context lengths, and the proportion of questions containing at least one shortcut edge. Table 6.2b then reports the main network characteristics of the underlying knowledge graph: its size (nodes and edges),

QA Metric	Mean	Std. Dev.
Question length (chars)	319.42	129.17
Question length (tokens)	45.49	18.64
Answer length (chars)	16.66	9.69
Answer length (tokens)	1.76	0.96
Mean # hops per question	2.45	1.12
Total context length (chars)	5993.10	5009.69
Total context length (tokens)	848.55	725.78
Hop length (chars, pooled)	2447.14	2222.35
Hop length (tokens, pooled)	346.49	324.56
Shortcut count per question	0.12	0.38
Hop-count Distribution (of 971 questions)		
1 hop	258	26.6%
2 hops	245	25.2%
3 hops	242	24.9%
4 hops	226	23.3%
5 hops	0	0%
Questions w/ 1 shortcut	96	(of 971)

(a) Dataset-level statistics for multi-hop questions

Graph Metric	Value
Number of nodes	14 523
Number of edges	13 419
Density	0.000127
Degree (min / max / avg)	0 / 257 / 1.85
Connected components	4 684
Largest component size	7 318
Avg. clustering coefficient	0.0298
Degree assortativity coefficient	–0.0265
Top 5 nodes by degree	
hydrogen (257), carbon (250), oxygen (232), CO_2 (220), lithium (155)	

(b) Key network-level properties of the loaded knowledge graph

Table 6.2: Overview of both dataset-level and graph-level statistics. Left: dataset stats for 971 questions; Right: key graph properties.

sparsity (density), degree distribution (min, max, and average), number of connected components and the size of the largest component, as well as clustering and assortativity coefficients. Finally, the five highest-degree nodes, hydrogen, carbon, oxygen, CO_2, and lithium, are listed to highlight the most central concepts in the graph.

Table 6.3 compares our dataset (ChemKGMultiHopQA) with HotpotQA-Chemistry and ChemLitQA-multi across question count, bridged entities, entity types, answer formats, domain, and source.

Dataset	# Qs	# bridged entities	# entity types	Answer type	Domain	Source
HotpotQA-Chemistry	980	0–4	General	Short	Wikipedia	Crowd (Wiki)
ChemLitQA-multi	742	1	Chemistry	Long & Short	ChemRxiv	LLM + expert verified
ChemKGMultiHopQA	971	0–3	Chemistry	Short	ChemRxiv + PubChem & Wikipedia	LLM + NER + KG (auto) + expert subset

Table 6.3: Comparison of HotpotQA, ChemLitQA-multi, and ChemKGMultiHopQA datasets.

6.4.2 Expert Feedback

We began with 52 multi-hop questions, each paired with a fully worked, hop-by-hop answer, and asked a panel of domain experts to rate answer quality. Twelve questions (23%) were dropped due to low evaluator confidence, leaving 40 high-confidence items. Table 6.4 categorizes these into Good (26, 65%), Ok (9, 22.5%), and Poor (5, 12.5%) based on expert ratings. For each category, we report the average number of models that correctly answered with full context versus none, along with the mean number of hops required.

Category	**Num. Qs**	**Avg. Correct (+ctx)**	**Avg. Correct (–ctx)**	**Avg. Hops**
Good	26 (65%)	7.5	4.2	2.46
Ok	9 (22.5%)	7.55	3.44	2.55
Poor	5 (12.5%)	7.8	4.6	2.60

Table 6.4: Expert-rated quality of 40 high-confidence questions. *Num. Qs* gives count and percentage; *Avg. Correct (+ctx)/(–ctx)* shows mean number of models answering correctly with and without context; *Avg. Hops* is the average reasoning hops.

6.4.3 Context and Reasoning

This analysis contrasts the performance of two model categories: *reasoning models*, which incorporate test-time reasoning, and *non-reasoning models*, which do not, under two conditions, one with contextual information and one without. As illustrated in Figure 6.5-A, the inclusion of context leads to a significant improvement in accuracy, nearly doubling the scores for both categories. Furthermore, reasoning models consistently outperform non-reasoning counterparts, gaining additional advantages from their reasoning abilities when context is available.

Consistent with expectations (Figure 6.5-B), non-reasoning models exhibit lower response times than reasoning models. For the latter, access to context decreases both latency and the number of output tokens, likely because the provided information reduces the need to construct complete reasoning chains from the ground up. Further details can be found in the Appendix.

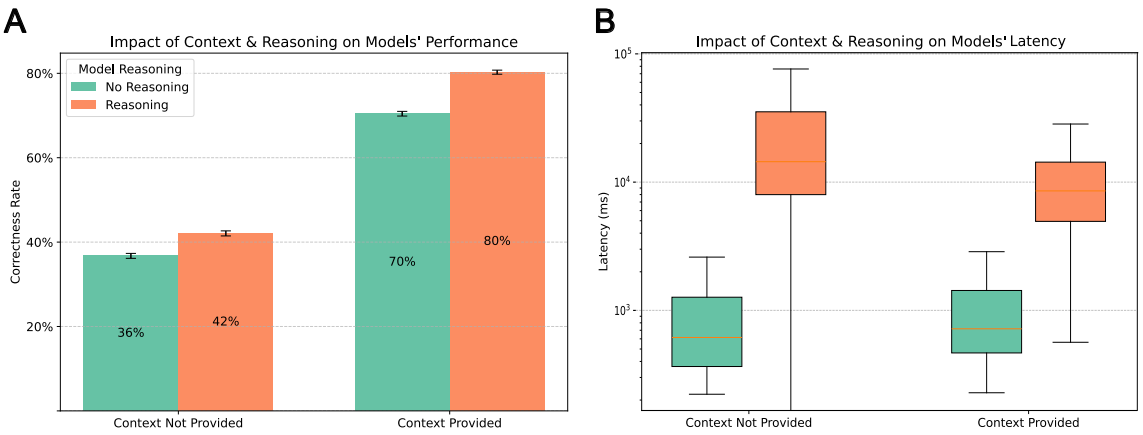


Figure 6.5: Impact of reasoning and context on model correctness rate (left) and latency (right). Error bars show the standard error of the mean across models within each category.

6.4.4 Impact of the Number of Hops

We next investigate how the number of reasoning hops affects correctness rate and output token count in the *context-provided* setup. Figure 6.6-A shows that as hop count increases, the number of tokens generated (i.e., reasoning steps) also rises, while the correctness rate for multi-hop questions remains relatively stable yet slightly lower than for single-hop tasks. For single-hop questions, higher token counts correlate with a small decrease in accuracy, suggesting that excessive reasoning may introduce noise

for simpler queries. These patterns vanish in the *no-context* setting (see Supplementary).

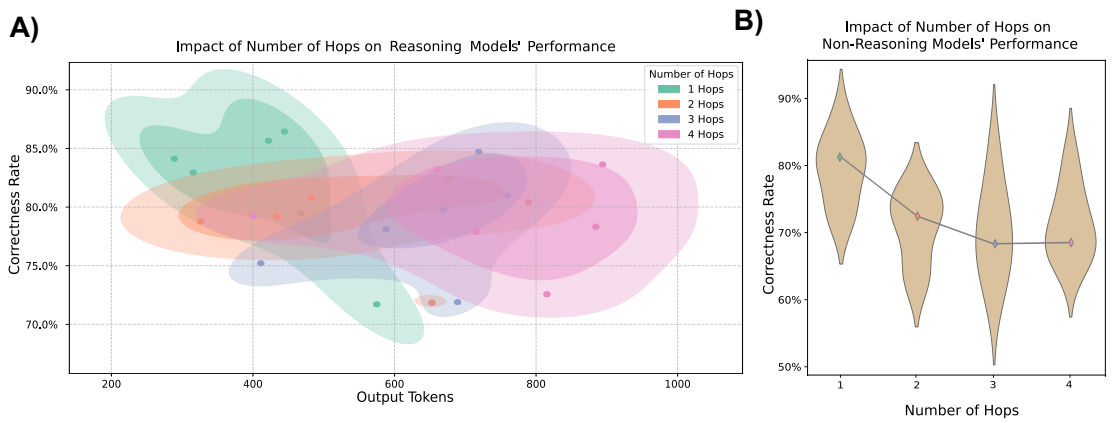


Figure 6.6: Analysis of hop count impact in the *context-provided* setup. **A**: Output token usage vs. correctness rate for reasoning models, colored by hop count. **B**: Correctness rate distributions for non-reasoning models across different hop counts (dots indicate medians).

6.5 Conclusion

In this study, we designed a domain-specific multi-hop question answering system and evaluated state of the art large language models in the chemistry domain. Our results show that these models have difficulty with in domain multi-hop scientific questions, answering fewer than half of the queries correctly when no context is provided. While models fine-tuned for reasoning demonstrate slightly better performance, they still encounter substantial challenges. Providing context leads to significant improvements, nearly doubling the performance of both reasoning and non-reasoning models. However, even with context, none of the models, including those fine-tuned for reasoning, achieved a perfect score.

We also introduce an automated pipeline that combines advanced named entity recognition with knowledge graph construction to produce complex multi-hop reasoning tasks for benchmarking. This framework, which is potentially domain agnostic, can be adapted to other fields by replacing the chemistry-specific named entity recognition with appropriate alternatives. It offers a strong foundation for future research aimed at enhancing reasoning abilities in specialized domains.

Limitations Our benchmark was evaluated in two setups: closed-book (no context) and open-book (full context). In real-world applications, context is typically retrieved incrementally during reasoning, suggesting a partial-context scenario that we did not explore. Non-reasoning models, in particular, might benefit from multi-step retrieval strategies rather than a single retrieval pass. Recent work on multi-hop RAG pipelines demonstrates promising approaches for iterative retrieval and answer refinement [174, 163, 214, 102]. Furthermore, chemically, evaluating a greater number of questions might be beneficial in checking the quality of work. Currently, only 40 questions were sampled and evaluated, which constitute about 4% of the generated questions.

Future Work Building on these findings, future work could investigate partial-context and incremental retrieval settings that more closely mimic real-world information access. Developing multi-hop RAG pipelines tailored to chemical texts, with iterative retrieval and reasoning cycles, may help bridge the performance gap observed in closed-book setups. Expanding the benchmark to include a larger and more diverse set of multi-hop questions, sourced from patents, research articles, and experimental reports, would improve its coverage and robustness. Additionally, exploring advanced prompting techniques and LLM chaining workflows, such as decomposing

complex queries into sub-questions and integrating intermediate reasoning steps, may further enhance large language models' ability to handle complex, in-domain multi-hop queries without requiring additional model training.

6.6 Attribution

Citation M. Khodadad, A. Shiraee Kasmaee, M. Astaraki, N. Sherck, H. Mahyar, and S. Samiee, "Evaluating Multi-Hop Reasoning in Large Language Models: A Chemistry-Centric Case Study," arXiv preprint arXiv:2504.16414, 2025. [Online]. Available: https://arxiv.org/abs/2504.16414 [79]

Contributions This work was developed through collaborative efforts. Mohammad Khodadad contributed to the ideation, preparation of the benchmark, preparation of the models, and writing. Ali Shiraee assisted with ideation and provided consultation. Mahdi Astaraki provided consultation and contributed to writing. Nicholas Sherck handled the chemical side of the work. Hamidreza Mahyar and Soheila Samiee supervised the project and provided guidance throughout.

Resources All preparations and experiments were conducted on internal BASF servers, utilizing 32 CPU cores and 64 GB RAM. The software environment included `Python`, `PyTorch`, `openai`, `aws`, and `git`, with package management handled via `pip`. Code is available at `https://zenodo.org/records/16882520` and `https://github.com/MohammadKhodadad/ChemKGMultiHopQA`

Chapter 7

Conclusion and Future Work

The rapid evolution of large language models (LLMs) and embedding techniques has revolutionized natural language processing, yet specialized domains such as chemistry and medicine present unique challenges that generic benchmarks and models often fail to address. This thesis tackled these challenges through three interrelated contributions: the development of domain-specific benchmarks (MedTEB and ChemTEB), the creation of a contrastively trained medical embedding model (MedTE), and the introduction of GraphRAG, an automated pipeline for multi-hop reasoning in chemistry. By systematically evaluating state-of-the-art models across these frameworks, we have advanced our understanding of domain adaptation, compositional reasoning, and evaluation methodologies in specialized NLP.

In Chapter 3, we introduced MedTE, a General Text Embedding (GTE) model fine-tuned via self-supervised contrastive learning on a richly curated corpus of PubMed abstracts, MIMIC-IV clinical notes, and other sources. Empirical results showed that MedTE consistently outperforms both general-purpose embeddings and prior medical models, demonstrating the decisive impact of contrastive objectives and diverse

domain data on embedding quality.

To fill the medical evaluation gap, Chapter 4 presented MedTEB, a benchmark suite of 51 tasks spanning classification, clustering, pair classification, and retrieval, tailored to the complexities of clinical and biomedical narratives. Complementing MedTEB,

In Chapter 5, we delivered ChemTEB, a novel benchmark for the chemical sciences that encompasses classification, bitext mining, clustering, retrieval, and molecular similarity tasks. ChemTEB's datasets, derived from PubChem, CoconutDB, Safety Data Sheets, and Wikipedia, were validated by chemists to ensure real-world relevance.

Building on these foundations, Chapter 6 introduced GraphRAG, an automated pipeline that constructs chemical knowledge graphs via NER and LLM-based relation extraction, then generates and validates multi-hop question–answer pairs. GraphRAG's evaluation of LLMs with and without RAG demonstrated that even perfect retrieval cannot fully mitigate reasoning errors in compositional tasks, highlighting fundamental limitations in current model architectures [192].

Looking ahead, several promising directions emerge:

- Improving MedTE by incorporating harder negatives (e.g., online and ontology-informed mining), expanding and continuously refreshing the training corpus with more real medical data (full-text clinical narratives, longitudinal records, multilingual sources), and applying lightweight domain-adaptive tuning to stay aligned with evolving terminology and low-resource subdomains.

- Extending MedTEB by adding bitext mining and reranking tasks, increasing coverage of real clinical data, introducing graded difficulty and structured/temporal

reasoning challenges, and injecting robustness evaluations (distribution shift, adversarial perturbations) alongside more realistic decision-support scenarios.

- Improving ChemTEB and developing a dedicated chemical embedding model by bolstering benchmark tasks, especially retrieval using domain-aware dense or hybrid methods, expanding task diversity to reflect modern chemist workflows, training a chemistry text model, and evaluating on ChemTEB.

- Refining GraphRAG by integrating retrieval into end-to-end evaluation instead of relying on golden context, enabling iterative multi-hop query reformulation, adding confidence and error detection mechanisms, dynamically updating and relevance-pruning the underlying knowledge graph, and incorporating expert-in-the-loop feedback.

In sum, this thesis demonstrates that domain-specialized benchmarks, contrastive pretraining, and automated reasoning pipelines are critical pillars for advancing NLP in chemistry and medicine. By open-sourcing our benchmarks, models, and pipelines, we aim to catalyze future research and drive the development of reliable, high-impact language technologies tailored to specialized scientific domains.

Appendix A

Appendix

A.1 MedTE

A.1.1 Loss

Figure A.1 presents the training and validation loss across the full pre-training schedule. While both curves show a steady decline, the average MedTEB score reaches its highest point around step 6000 before dropping by step 8000. This difference occurs because decreases in contrastive loss do not necessarily correspond to better downstream performance, particularly when benchmark data is excluded from both training and validation sets. In other words, even if contrastive loss improves, downstream evaluation is still crucial to confirm actual performance gains.

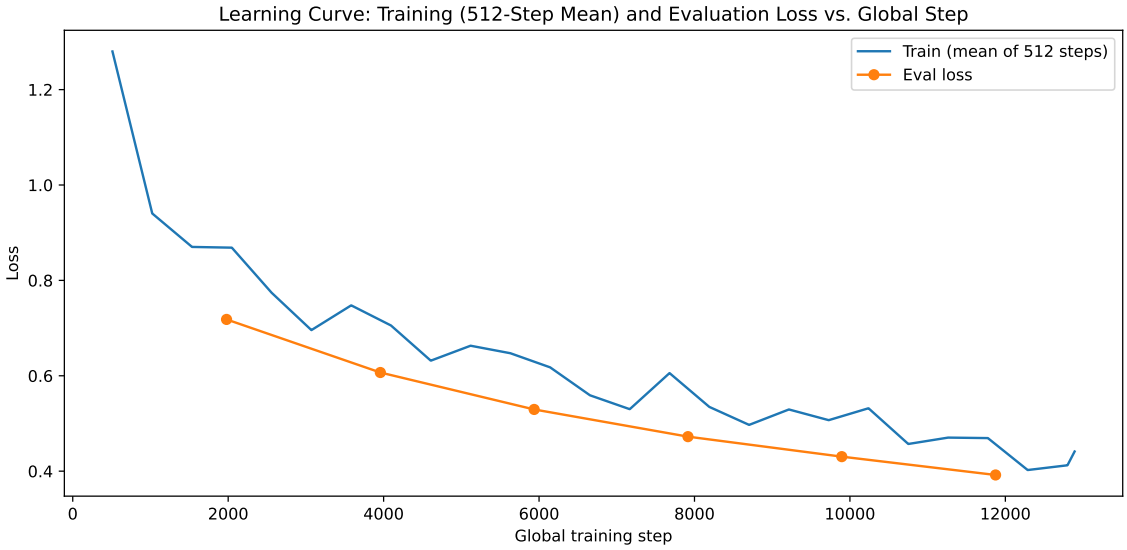


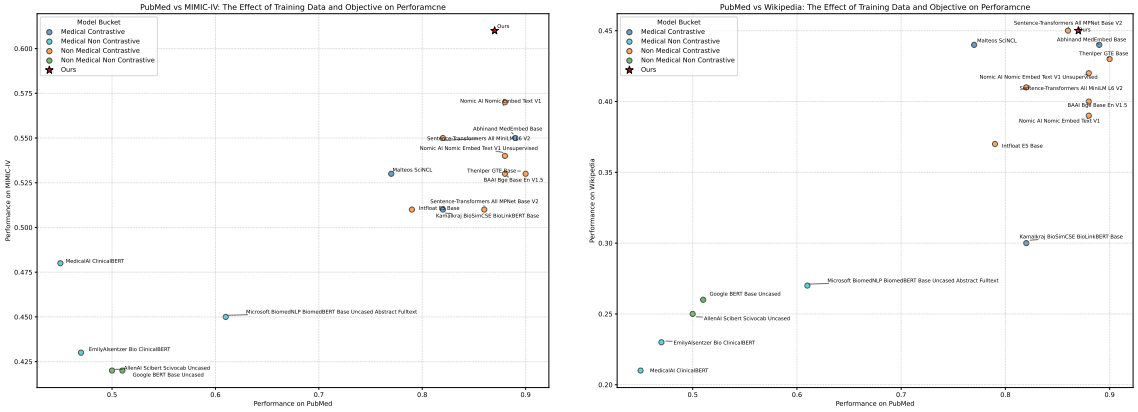
Figure A.1: Training and validation loss curves for MedTE.

A.2 MedTEB

A.2.1 wiki-pubmed and pubmed-mimic-IV

In Figure 4.1, the results illustrate how the assessed models perform on MIMIC-IV compared with Wikipedia, revealing a pronounced disparity between clinical language and general-domain content. Figures A.2a and A.2b extend this analysis by incorporating PubMed, a large, peer-reviewed biomedical literature repository, into the evaluation, enabling direct comparison across MIMIC-IV, Wikipedia, and PubMed. By adding this third corpus, we can more clearly investigate the impact of exposure to formal biomedical writing on cross-domain generalization. Models that undergo domain-specific pretraining on PubMed tend to narrow the gap in MIMIC-IV performance while retaining strong results on Wikipedia, demonstrating both the

strengths and the inherent trade-offs of leveraging specialized corpora for tasks spanning biomedical and general language contexts.

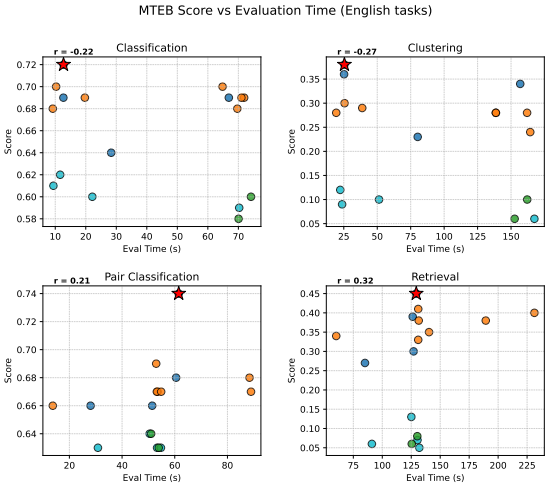


(a) PubMed vs MIMIC-IV (b) PubMed vs Wikipedia

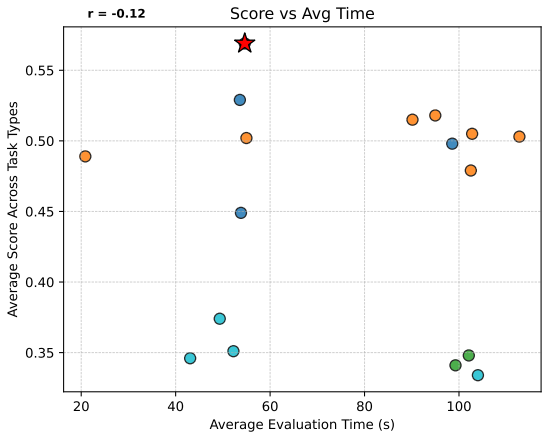
Figure A.2: Comparison of document styles across datasets.

A.2.2 Runtime–effectiveness overview

Figure A.3a compares *evaluation time* (in seconds) with *task specific effectiveness* across the four benchmark families: classification, clustering, pair classification, and retrieval. In each scatter plot, our model appears in the upper left region, indicating both high accuracy and low latency. Pearson correlation values range from $r = -0.32$ for retrieval to $r = -0.07$ for clustering, suggesting only a weak and sometimes negative relationship between longer evaluation times and higher scores. Figure A.3b presents the same data averaged per model, showing a similarly small overall trend ($r = -0.12$) and reinforcing that efficient contrastive training can deliver state of the art performance without increasing inference cost.



(a) Evaluation time vs. effectiveness across all task categories



(b) Per-model average performance and evaluation time

(c) Legend for the four model groups

Figure A.3: (a) Evaluation time vs. effectiveness across all task categories; (b) Per-model average performance vs. evaluation time; (c) Legend for the four model groups.

A.3 Chemteb

A.3.1 Models Spec

	Model Name	HuggingFace Model / Model ID (Proprietary)	Model Size	# Parameters	Context length	Embedding size
	Open-Source Models					
1	BERT	google-bert/bert-base-uncased	440 MB	109.4 M	512	768
2	SciBERT	allenai/scibert_scivocab_uncased	442 MB	109.9 M	512	768
3	MatSciBERT	m3rg-iitd/matscibert	440 MB	109.9 M	512	768
4	Chemical BERT	recobo/chemical-bert-uncased	440 MB	109.9 M	512	768
5	Nomic BERT	nomic-ai/nomic-bert-2048	549 MB	136.7 M	2048	768
6	Nomic Embedding v1	nomic-ai/nomic-embed-text-v1	547 MB	136.7 M	8192	768
7	Nomic Embedding v1.5	nomic-ai/nomic-embed-text-v1.5	547 MB	136.7 M	8192	768
8	SBERT - all Mini LM L6.v2	sentence-transformers/all-MiniLM-L6-v2	90.9 MB	22.7 M	512	384
9	SBERT - all Mini LM L12.v2	sentence-transformers/all-MiniLM-L12-v2	133 MB	33.3 M	512	384
10	SBERT - all MPNET-base.v2	sentence-transformers/all-mpnet-base-v2	438 MB	109.4 M	514	768
11	SBERT - multi-qa-mpnet-base.v1	sentence-transformers/multi-qa-mpnet-base-dot-v1	438 MB	109.4 M	512	768
12	E5 - small	intfloat/e5-small	133 MB	33.3 M	512	384
13	E5 - base	intfloat/e5-base	438 MB	109.4 M	512	768
14	E5 - large	intfloat/e5-large	1.34 GB	335.1 M	512	1024
15	E5 - small v2	intfloat/e5-small-v2	133 MB	33.6 M	512	384
16	E5 - base v2	intfloat/e5-base-v2	438 MB	109.4 M	512	768
17	E5 - large v2	intfloat/e5-large-v2	1.34 GB	335.1 M	512	1024
18	E5 - Multilingual small	intfloat/multilingual-e5-small	471 MB	117.6 M	512	384
19	E5 - Multilingual base	intfloat/multilingual-e5-base	1.11 GB	278 M	514	768
20	E5 - Multilingual large	intfloat/multilingual-e5-large	2.24 GB	559.8 M	514	1024
21	BGE - small en	BAAI/bge-small-en	133 MB	33.3 M	512	384
22	BGE - base en	BAAI/bge-base-en	438 MB	109.4 M	512	768
23	BGE - large en	BAAI/bge-large-en	1.34 GB	335.1 M	512	1024
24	BGE - small en v1.5	BAAI/bge-small-en-v1.5	133 MB	33.3 M	512	384
25	BGE - base en v1.5	BAAI/bge-base-en-v1.5	438 MB	109.4 M	512	768
26	BGE - large en v1.5	BAAI/bge-large-en-v1.5	1.34 GB	335.1 M	512	1024
27	BGE - Multilingual - M3	BAAI/bge-m3	2.27 GB	576.7 M	8192	1024
	Proprietary Models					
28	OpenAI - Text embedding 3 - small	text-embedding-3-small	N/A	N/A	8191	1536
29	OpenAI - Text embedding 3 - large	text-embedding-3-large	N/A	N/A	8191	3072
30	OpenAI - Text embedding - Ada - 02	text-embedding-ada-002	N/A	N/A	8191	1536
31	Amazon - Titan Text Embedding v2	amazon.titan-embed-text-v2:0	N/A	N/A	8191	1536
32	Amazon - Titan Embedding G1 Text	amazon.titan-embed-text-v1	N/A	N/A	8191	1536
33	Cohere - Embed English V3	cohere.embed-english-v3	N/A	N/A	512	1024
34	Cohere - Embed Multilingual V3	cohere.embed-multilingual-v3	N/A	N/A	512	1024

Table A.1: This table summarizes the embedding models, highlighting each model's name, HuggingFace model or proprietary ID, model size on disk, number of parameters, the maximum context length, and the default embedding dimension. Models are categorized into open-source and proprietary sections for easier distinction.

A.3.2 Ranking of models

Detailed ranking of models on each category of tasks is provided in Table A.2. The ranking is calculated based on average performance over all tasks in each category defined on them.

Table A.2: Summary of models rank

	Classification	Bitext Mining	Retrieval	Clustering	Pair Classification	RRF_Score(k=10)
Nomic BERT	34	33	34	27	34	0.118
Chemical BERT	33	30	32	33	31	0.120
MatSciBERT	32	31	33	28	32	0.122
BERT	30	34	30	29	33	0.122
SciBERT	31	32	31	31	30	0.122
BGE - small en	12	27	26	25	22	0.160
E5 - small v2	23	25	8	30	29	0.165
E5 - small	25	21	9	34	24	0.166
BGE - Multilingual - M3	21	26	10	15	28	0.176
E5 - base v2	22	18	11	19	25	0.178
BGE - small en v1.5	9	23	18	26	20	0.180
SBERT - multi-qa-mpnet-base.v1	28	29	24	17	5	0.185
BGE - base en	16	13	22	16	19	0.186
E5 - Multilingual large	27	7	14	24	23	0.187
BGE - large en	10	19	29	13	17	0.191
E5 - base	20	14	12	22	15	0.192
E5 - Multilingual base	26	11	13	10	27	0.196
SBERT - all Mini LM L12.v2	18	22	23	21	4	0.201
E5 - Multilingual small	29	16	1	32	26	0.207
E5 - large v2	24	12	3	23	21	0.214
BGE - base en v1.5	15	17	7	11	18	0.219
BGE - large en v1.5	6	15	15	18	12	0.224
Amazon - Titan Text Embedding v2	17	8	19	8	14	0.224
SBERT - all Mini LM L6.v2	8	20	20	20	3	0.232
SBERT - all MPNET-base.v2	7	28	25	6	6	0.239
OpenAI - Text embedding 3 - small	5	5	17	9	10	0.273
Cohere - Embed English V3	2	24	28	2	8	0.278
OpenAI - Text embedding - Ada - 02	11	2	16	4	16	0.279
Cohere - Embed Multilingual V3	4	9	27	3	9	0.281
Nomic Embedding v1	19	10	4	12	2	0.285
Amazon - Titan Embedding G1 Text	1	3	21	14	13	0.285
E5 - large	14	4	6	5	11	0.290
Nomic Embedding v1.5	13	6	2	7	1	0.339
OpenAI - Text embedding 3 - large	3	1	5	1	7	0.384

A.3.3 Comparison of MTEB and ChemTEB

Figure A.4 reflects the performance of each model in each category of tasks on both benchmarks. In three out of four categories of tasks, the BERT model provided the weakest performance in both benchmarks.

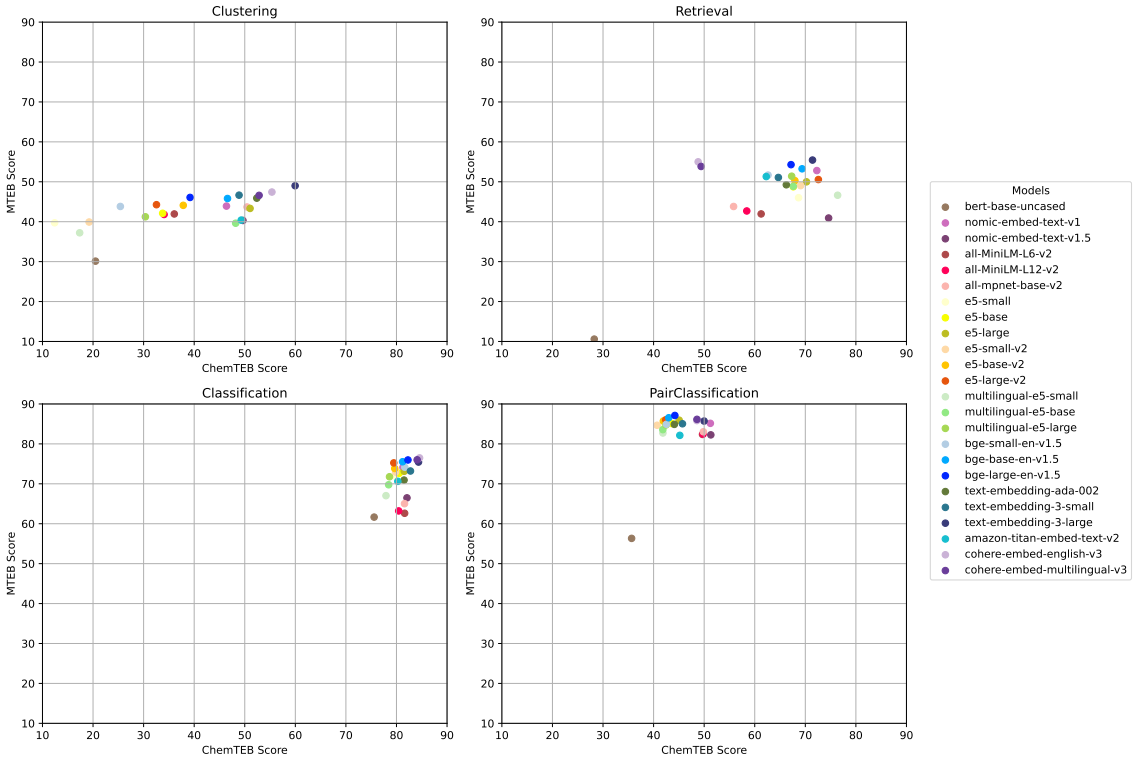


Figure A.4: Comparison of models' performance on ChemTEB and MTEB benchmarks across different tasks. Each point represents a model from the intersection of those tested and those on the MTEB leaderboard as of the date. The figure highlights variations in task difficulty and domain specificity.

A.3.4 Correlation between models performances and tasks

Figures A.5 and A.6 illustrate the correlation matrix for the datasets and models, respectively, with colors representing the strength of the correlations. In figure A.5, We can observe that in tasks such as classification, bitext mining, and retrieval, the datasets in a task are correlated except for the SDS datasets in the classification. In the clustering, and pair classification task, however, this trend is not very obvious. Especially, in the pair classification task, some of the datasets have negative correlation.

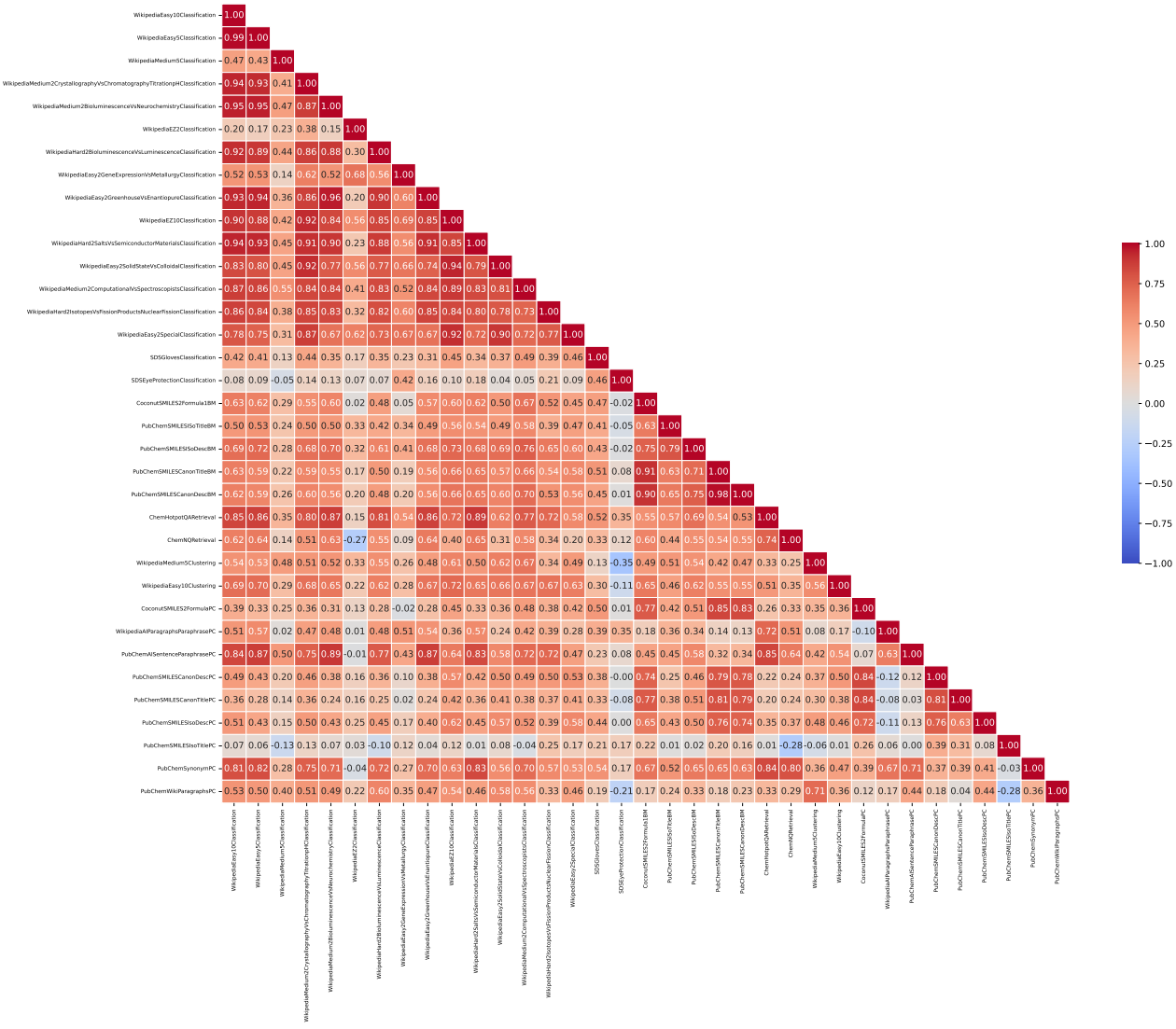


Figure A.5: Correlation Matrix across datasets. Each row and column represents a separate dataset tested in the ChemTEB benchmark. The values and associated color reflect the correlation between the performance of different models on each pair of these datasets.

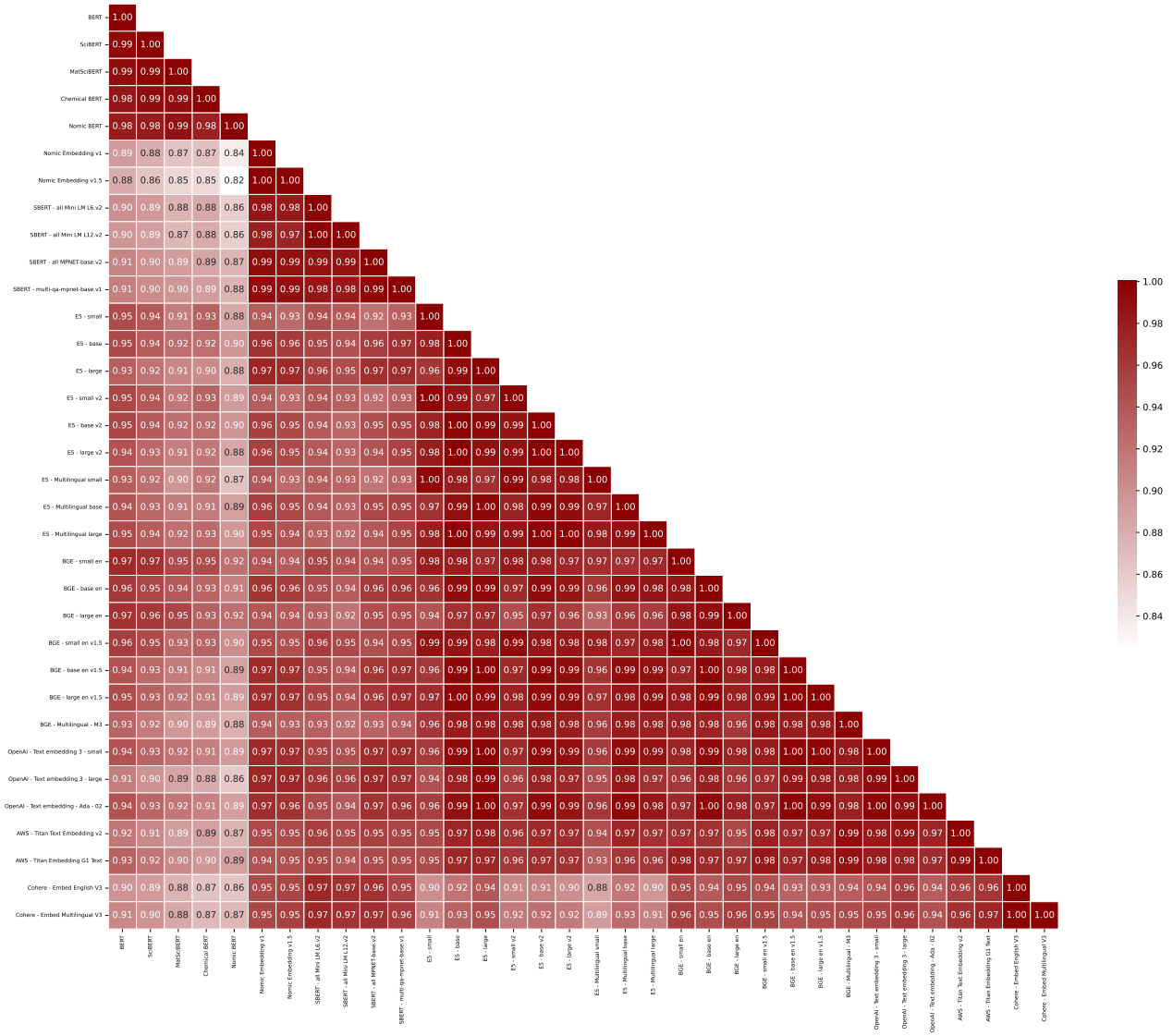


Figure A.6: Correlation Matrix over Models. Each row and column represents a separate Model tested in the ChemTEB benchmark. The values and associated color reflect the correlation between the performance of each pair of models over all tested datasets.

A.4 Multi-hop QA over Graph

A.4.1 Detailed Knowledge Graph Generation

In this section, we explain each step in our graph generation process, illustrating how unstructured chemical text is transformed into a structured representation suitable for downstream tasks.

Text Preprocessing

We begin by fetching all ChemRxiv articles licensed for redistribution and extracting each introduction via regex. From each introduction, we take the first 500 words, enough to capture key background while avoiding noise, making sure not to split paragraphs. We then split each 500-word excerpt into contiguous, sentence-bounded chunks of up to 128 words, keeping each piece below our downstream models' token limits.

Node Extraction

For each 128-word chunk, we run a PubMedBERT-based NER model (fine-tuned on chemical corpora) to propose entity spans. We then prompt GPT-4o with a few-shot task to validate each span as a true chemical entity and normalize labels (e.g., "MeOH" $\rightarrow$ "methanol").

> You are a chemistry expert specializing in entity recognition. Your task is to **validate and filter** the extracted entities, ensuring they are **chemically meaningful** based on the provided text. Remove any irrelevant terms, including general

descriptors, numerical values, reaction conditions, and vague terms.

Entities Extracted by NER:

{entities}

Text for Context:

{text}

Criteria for Valid Entities:

✓ Chemical compounds (e.g., "HCl", "Sodium hydroxide", "Ethanol", "Benzene")

✓ Chemical elements (e.g., "Carbon", "Oxygen", "Cesium")

✓ Specific catalysts, solvents, reagents (e.g., "Cs_2CO_3", "Toluene", "Palladium")

Remove the Following Types of Entities:

× Generic terms (e.g., "Reaction", "Solvent", "Acid", "Base", "Solution")

× Experimental conditions (e.g., "pH", "Temperature", "2 M", "Strong acid")

× Measurement terms (e.g., "X-ray diffraction", "NMR")

× General descriptors (e.g., "High concentration", "Low efficiency")

Output Format:

Return only a **Python list** of valid chemical entities, with no explanations, markdown, or extra formatting.

Edge Extraction

Using the vetted entities, we prompt the same `gpt-4o` model on each co-occurring pair to classify or generate their relationship, yielding triplets (entity_A, relation, entity_B). This produces a graph of chemical nodes and directed edges that capture functional associations from the text. Below is the edge-extraction prompt.

> You are an expert in chemical text analysis. Your task is to extract **only chemically meaningful relationships** between a given set of entities from the provided text.
>
> **Guidelines for Relation Extraction:**
>
> 1. **Entity Matching:** Consider only the entities provided in the given set. If an entity appears in the text but has no meaningful chemical relationship with another entity in the set, ignore it.
> 2. **Chemically Significant Relations Only:** Extract relations that describe actual **chemical interactions, transformations, or properties** (e.g., "reacts with," "catalyzes," "dissolves in," "produces").
> 3. **Factual Relations:** Only extract factual relations. Avoid observations, opinions, and findings.
> 4. **Tuple Format:** Output extracted facts in the form of **(entity1, relation, entity2)**.
> 5. **Avoid Generic Relations:** Exclude weak relations like "is," "are," "exists," "relates to." Focus on **specific interactions**.

Valid Relation Types (Examples):

✓ "reacts with"

✓ "catalyzes"

✓ "binds to"

✓ "dissolves in"

✓ "oxidizes"

✓ "inhibits"

✓ "precipitates with"

✓ "acts as a solvent for"

✓ "is synthesized from"

Avoid These Weak Relations: Exclude relations such as "is," "are," "has," "exists."

Entities Provided:

{`entities`}

Text:

{`text`}

Extract at most {`max_facts`} factual statements.

Output Format:

Provide the output as a **Python list of tuples**, containing only the extracted relationships without any code formatting, backticks, or markdown.

Example Output:

– [(“HCl”, “dissolves in”, “Water”), (“HCl”, “reacts with”, “Sodium hydroxide”)]

Knowledge Enrichment

We enhance each node by fetching its Wikipedia summary and PubChem data, official name, alternate identifiers, record description, safety annotations, canonical SMILES, molecular formula, and key physicochemical properties (e.g., molecular weight, TPSA, logP), and store these as external metadata.

Graph Generation

We build the knowledge graph as triplets (node, edge, node), linking each element to its source text and any retrieved metadata to maintain full traceability.

A.4.2 Detailed Question Generation

Path Sampling from the Knowledge Graph

Using randomized BFS, we sample paths of length K (with $K + 1$ entities and K edges), enforcing that each edge comes from a distinct source document to require multi-document reasoning.

One-Hop Question Formulation

For each triplet (e_1, r, e_2), we generate a question, "Which entity has the r relation to e_2?", and, if necessary, enrich the prompt with contextual metadata via an LLM to ensure clarity and answerability from the original text.

You are given a text along with an entity and its relation to another entity.

Entity 1: {`entity1`}
Relation: {`relation`}
Entity 2: {`entity2`}
Text: {`text`}

Information about Entity1: {`entity1_meta` if `entity1_meta` else `None`}

Your task is to generate a factual question whose answer is Entity1.
The question should ask for the entity that has the specified relation to Entity2.
Do not mention the answer (which is Entity1) in the question.
Ensure that the question is factual and can be answered solely based on the given text and the information about Entity1.
Do not refer to sections such as "Abstract," "Table 1," "in the text," or "in the article."

If Entity1 and relation are not specific enough (i.e., multiple answers are possible), add descriptions from the text or from the information about Entity1

to make it specific so that Entity1 is the only answer.

Return a dictionary without any code formatting, backticks, or markdown, with keys `\q"` and `\a"`.

Multi-Hop Question Aggregation

We merge vetted one-hop prompts into a single multi-hop question via few-shot prompting on `o3-mini`. Beginning with the sub-question for entity$_{K+1}$, we chain backward through each relation to entity$_1$. This reverse-chaining guarantees logical flow and targets entity$_1$ as the final answer. Below is the multi-hop question prompt.

You are given multiple factual questions and their answers that are logically connected.
Your task is to chain them into a single, coherent multi-hop question that requires multiple reasoning steps.
Ensure that the (only) answer is the answer to the first question, and the question naturally follows from the facts given.
You have to start from the last generated question and build up a single multi-hop question so it aggregates them all and the answer is the answer to the first question.
None of the answers to any of the questions should be in the generated question.

Here is an example:

Example:

Q1: What is oxidized to form Carbon Dioxide?

A1: Methane

Q2: What is used in Photosynthesis?

A2: Carbon Dioxide

Q3: What produces Oxygen?

A3: Photosynthesis

Multi-hop question:

Q: What is oxidized to produce a substance that is used in a process that results in Oxygen?

A: Methane

Here are the generated questions and answers:

{`formatted_qas`}

Return a python dictionary without any code formatting, backticks, or markdown, with keys `"q"` (multi-hop question) and `"a"` (final answer).

Verification and Filtering

We validate each one-hop question with a few-shot chemistry-expert prompt, ensuring it is factual, unambiguous, and directly answerable from its context. Below is the evaluation prompt:

You are a chemistry expert. Your task is to determine if the given question is a factual chemistry question, unambiguous (has only one answer), and answerable based on the provided context. A factual question must be based on actual chemical properties, reactions, or experimentally verified principles and must be strictly related to chemistry. An answerable question should be solvable based on the given context and must not be open-ended or have multiple correct answers. MAKE SURE THE QUESTION HAS ONLY ONE CORRECT ANSWER. There shouldn't be any other entity except for the given answer that could be another answer.

Question:

{question}

Answer:

{answer}

Context:

{context}

Please analyze the context and verify if the question is factual, unambiguous, and answerable. If the question is factual, has only one correct answer, is strictly related to chemistry, and can be answered based on the context, return "yes." Otherwise, return "no."

Examples of Factual Chemistry Questions:

"What dissolves in water and evaporates at 0 °C?"

"What catalyst is used in the reaction between A and B?"

Examples of Non-Factual or Ambiguous Chemistry Questions:

"What is the song of Nirvana that is a chemical entity?"

"What chemical entity and structural unit form the layered hydroxide structures with intercalated water ions used in battery materials and OER catalysis?" ($M(OH)_6$ and -$Ni(OH)_2$ are valid answers)

Questions that have multiple possible correct answers or are not strictly related to chemistry.

We also validate the entire multi-hop chain with an LLM-based prompt and expert feedback, ensuring each sub-question leads coherently and factually to entity_1 using available context and metadata. Prompts are refined iteratively, and any question misanswered by all models is discarded. Below is the path-evaluation prompt.

You are a chemistry expert. Your task is to determine if the given question is a factual chemistry question and answerable based on the provided path.

Path Information:

{`path_text`}

Question:

{question}

Answer:

{answer}

Please analyze the path and verify if the question is a factual chemistry question and can be answered based on the given path. A factual question must be based on actual chemical properties, reactions, or experimentally verified principles. An answerable question should be solvable based on the given path. If the question is factual and answerable, return "yes". If it contains speculation, opinions, or lacks verifiable chemical grounding, or it is not solvable, return "no".

Examples of Factual Chemistry Questions:

"What dissolves in water?"

"What catalyst is used in the reaction between A and B?"

"Which compound undergoes oxidation in this reaction?"

"What product is formed when sodium reacts with chlorine?"

Examples of Non-Factual Chemistry Questions:

"Why do some scientists think this reaction is inefficient?"

"What is the best solvent for this reaction?"

"Is this reaction useful in industry?"

"Do you think this compound is a good catalyst?"

Provide only “yes” or “no” as your response.

A.4.3 Rejected Questions

We excluded questions that (1) admit multiple valid answers or (2) include the answer verbatim in the question. Examples are shown below.

Q1:

Context:

Researchers have developed an anode material based on $NiCo_rGO$ (Nickel–Cobalt–reduced Graphene Oxide). In some variants, the $NiCo_rGO$ is further decorated with palladium (Pd) nanoparticles to enhance catalytic performance.

Question:

Which component in the electrode structure functions as a catalyst at the anode when incorporating decorated $NiCo_rGO$?

Issue:

The question is declined due to ambiguity , two distinct answers are technically correct based on the variant of the material:

- If the material is **$NiCo_rGO$** without Pd, the catalyst is **Nickel (Ni)**.
- If the material is **Pd-decorated $NiCo_rGO$**, the catalyst is **Palladium (Pd)**.

Since the phrasing of the question does not clearly disambiguate which material variant is being used, it leads to multiple valid interpretations. Therefore, it cannot be accepted as a single-answer question.

Q2:

Context:

Hypervalent iodine compounds such as diaryliodonium salts are widely used as electrophilic arylation reagents. According to the source text, these salts are employed in both **transition metal-catalyzed** and **metal-free** arylation reactions. These reactions can be used to functionalize aromatic compounds, including halogen-substituted analogues like CIMPPC, by replacing hydrogen atoms.

Question:

Which type of arylation reaction that employs electrophilic arylation reagents utilizes diaryliodonium salts in hypervalent iodine chemistry?

Expected Answer:

Transition metal-catalyzed arylations

Reason for Rejection:

The question is declined due to the presence of **multiple valid answers**. The source explicitly states that diaryliodonium salts are used in:

- **Transition metal-catalyzed arylations**, and
- **Metal-free arylations**.

Both are equally valid interpretations of the question. Without further constraints or clarification, the question has more than one correct answer and does not meet

the single-answer requirement.

We removed any questions misanswered by all models or flagged by the overall LLM verifier (Section 6.2.2); manual review showed most discarded items had multiple valid answers.

A.4.4 A Multi-Hop QA Generation Example

Figure A.7 presents a multi-hop QA example: carbon dioxide is converted to formic acid, which then acts as a CO surrogate in carbonylation. Chaining these steps produces a question that integrates multiple facts. This showcases how entity relations and metadata enable complex QA generation and LLM evaluation. Figure 6.2 outlines the full graph-to-question pipeline.

Context:

[Source 1*]: Carbonylation reactions constitute a potent tool to manufacture carboxylic acids and their derivatives both in industry and academic organic synthesis. In general, carbonylation requires the use of toxic carbon monoxide, which thus usually demands certified high-pressure reaction vessels. Therefore, developing non-gaseous CO surrogate for conducting safe and facile-operation carbonylation is an important and ongoing research topic. Among these established CO surrogates, formic acid is one kind of versatile atom.

[Source 2*]: The utilization of carbon dioxide as a C1 feedstock for the generation of industrially relevant chemicals is also an interesting approach. CO_2 is an attractive renewable C1 source, which can lead to formic acid. Those approaches would not only reduce carbon dioxide emissions through carbon capture but also compensate sequestration costs by producing chemicals in global demand.

Question:
What is the process that uses a substance, produced from carbon dioxide and known as the simplest carboxylic acid with antibacterial and preservative properties, as a non-gaseous surrogate to safely form carboxylic acids and their derivatives under mild conditions?

Answer: carbonylation reactions

Sentence-level supporting facts:
1) formic acid can be produced from carbon dioxide.
2) formic acid is the simplest carboxylic acid with antibacterial and preservative properties.
3) formic acid can act as a non-gaseous CO surrogate.
4) carbonylation reactions safely produce carboxylic acids under mild conditions using formic acid as a CO surrogate.

Path (multi-hop chain of reasoning):
carbon dioxide → formic acid → carbonylation reactions

* Source 1 and source 2 are coming from different documents.

Figure A.7: An example of a multi-hop question-answer. [79]

Bibliography

[1] W. A. Abro, H. Kteich, and Z. Bouraoui. Self-supervised segment contrastive learning for medical document representation. In *International Conference on Artificial Intelligence in Medicine*, pages 312–321. Springer, 2024.

[2] A. Akbik, D. Blythe, and R. Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 1638–1649, 2018.

[3] O. K. Alhmoudi, M. Aboushanab, M. Thameem, A. Elkamel, and A. A. Al-Hammadi. Domain adaptation of a SMILES chemical transformer to SELFIES with limited computational resources. *Scientific Reports*, 15:23627, 2025. doi: 10.1038/s41598-025-05017-w.

[4] M. Alkhalaf, P. Yu, M. Yin, and C. Deng. Applying generative ai with retrieval augmented generation to summarize and extract key clinical information from electronic health records. *J. Biomed. Inform.*, 156:104662, 2024. doi: 10.1016/j.jbi.2024.104662.

[5] R. AlSaad, A. Abd-alrazaq, S. Boughorbel, A. Ahmed, M.-A. Renault, R. Damseh, and J. Sheikh. Multimodal large language models in health care:

Applications, challenges, and future outlook. *Journal of Medical Internet Research*, 26:e59505, 2024. doi: 10.2196/59505.

[6] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, 2019.

[7] D. Araci. FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[8] V. Bagal, R. Aggarwal, P. K. Vinod, and U. D. Priyakumar. Molgpt: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62(9):2064–2076, 2022.

[9] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2015.

[10] D. Bajusz, A. R'acz, and K. H'eberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1):20, 2015.

[11] A. Balachandran. Medembed: Medical-focused embedding models, 2024. URL `https://github.com/abhinand5/MedEmbed`.

[12] G. Balde, S. Roy, M. Mondal, and N. Ganguly. Evaluation of llms in medical text summarization: The role of vocabulary adaptation in high oov settings. In *Findings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*, 2025.

[13] M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 238–247, 2014.

[14] J.-C. B'elisle-Pipon. Why we need to be careful with llms in medicine. *Frontiers in Medicine*, 11:1495582, 2024. doi: 10.3389/fmed.2024.1495582.

[15] I. Beltagy, K. Lo, and A. Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of EMNLP*, pages 3613–3618, 2019.

[16] A. Ben Abacha and D. Demner-Fushman. A question-entailment approach to question answering. *BMC Bioinform.*, 20(1):511:1–511:23, 2019. URL `https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3119-4`.

[17] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181.

[18] O. Bodenreider. The unified medical language system (umls): Integrating biomedical terminology. *Nucleic Acids Research*, $32(\text{suppl}_1) : D267 - -D270, 2004$.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[20] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. Bernstein, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[21] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[22] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[23] K. Bostrom and G. Durrett. Byte pair encoding is suboptimal for language model pretraining. In *Findings of EMNLP*, pages 4617–4624, 2020.

[24] A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White, and P. Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6:525–535, 2024.

[25] N. Brown, M. Fiscato, M. H. S. Segler, and A. C. Vaucher. Guacamol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019. doi: 10.1021/acs.jcim.8b00839.

[26] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[27] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, P. Shyam, G. Sastry, A. Askell, S. Agarwal, et al. Language models are few-shot learners. In

Advances in Neural Information Processing Systems 33 (NeurIPS 2020), pages 1877–1901, 2020.

[28] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, 2017.

[29] P. Chandak, K. Huang, and M. Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67, 2023.

[30] D. Chang, I. Balažević, C. Allen, D. Chawla, C. Brandt, and A. Taylor. Benchmark and best practices for biomedical knowledge graph embeddings. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 167–176, 2020.

[31] H. Chen, L. Dan, Y. Lu, M. Chen, and J. Zhang. An improved data augmentation approach and its application in medical named entity recognition. *BMC Medical Informatics and Decision Making*, 24:221, 2024. doi: 10.1186/s12911-024-02624-x.

[32] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu. M3–Embedding: Multi–linguality, multi–functionality, multi–granularity text embeddings through self–knowledge distillation. *arXiv preprint*, arXiv:2402.03216, 2024. URL `https://arxiv.org/abs/2402.03216`.

[33] R. Chen, W. Jiang, C. Qin, I. S. Rawal, C. Tan, D. Choi, B. Xiong, and B. Ai. Llm-based multi-hop question answering with knowledge graph integration in evolving environments. *arXiv preprint arXiv:2408.15903*, 2024.

[34] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179.

[35] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. F. Stewart. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pages 3504–3512, 2016.

[36] A. Chowdhery, S. Narang, J. Devlin, B. Zoph, W. Chen, A. Roberts, , et al. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2202.08906*, 2022.

[37] Clinia Newsroom. Introducing heMTEB, an open-source benchmark for health information retrieval. `https://clinia.com/en-ca/newsroom/introducing-hemteb-an-open-source-benchmark-for-health-information-retrieval`, 2024.

[38] A. Conneau and D. Kiela. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018. ELRA.

[39] G. V. Cormack, C. L. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759, 2009.

[40] R. Cornet and N. de Keizer. Forty years of snomed: a literature review. *BMC medical informatics and decision making*, 8(Suppl 1):S2, 2008.

[41] A. Creswell, M. Shanahan, and I. Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*, 2022.

[42] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[43] J. Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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 (NAACL-HLT)*, pages 4171–4186, 2019.

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[46] J. R. Firth. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*, 1957, 1957.

[47] D. Garay-Ruiz and C. Bo. Chemical reaction network knowledge graphs: the OntoRXN ontology. *Journal of Cheminformatics*, 14:29, 2022. doi: 10.1186/s13321-022-00610-x.

[48] O. K. Gargari and G. Habibi. Enhancing medical ai with retrieval-augmented generation: A mini narrative review. *Digital Health*, 11:1–11, 2025. doi: 10.1177/20552076251337177.

[49] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. Domain-specific language model pretraining for biomedical natural language processing, 2020.

[50] P. Gupta and M. Jaggi. Obtaining better static word embeddings using contextual embedding models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL 2021)*, pages 5241–5253, 2021.

[51] T. Gupta, M. Zaki, N. A. Krishnan, and Mausam. Matscibert: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):102, 2022.

[52] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[53] S. Gururangan, A. Marasovi'c, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*, pages 8342–8360, 2020.

[54] B. J. Gutierrez, H. Sun, and Y. Su. Biomedical Language Models are Robust to Sub-optimal Tokenization. *arXiv preprint arXiv:2306.17649*, 2023.

[55] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

[56] H. Han, Y. Wang, H. Shomer, K. Guo, J. Ding, Y. Lei, M. Halappanavar, R. A. Rossi, S. Mukherjee, X. Tang, Q. He, Z. Hua, B. Long, T. Zhao, N. Shah, A. Javari, Y. Xia, and J. Tang. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*, 2025.

[57] Y. Han, Z. Wan, L. Chen, K. Yu, and X. Chen. From Generalist to Specialist: A Survey of Large Language Models for Chemistry. *arXiv preprint arXiv:2412.19994*, 2024.

[58] Y. He, X. Zhu, D. Li, and H. Wang. Enhancing Large Language Models for Specialized Domains: A Two-Stage Framework with Parameter-Sensitive LoRA Fine-Tuning and Chain-of-Thought RAG. *Electronics*, 14(10):1961, 2025. doi: 10.3390/electronics14101961.

[59] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[60] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, 2019.

[61] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 328–339, 2018.

[62] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 328–339, 2018.

[63] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[64] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[65] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 4411–4421, 2020.

[66] D. Huang and J. M. Cole. Cost-Efficient Domain-Adaptive Pretraining of Language Models for Optoelectronics Applications. *J. Chem. Inf. Model.*, 65:2476–2486, 2025. doi: 10.1021/acs.jcim.4c02029.

[67] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 873–882, 2012.

[68] G. Huang, Y. Long, Y. Li, G. Papanastasiou, G. Skaltsas, I. Riga, H. Weerts, N. Korfiatis, et al. From explainable to interpretable deep learning for natural language

processing in healthcare: How far from reality? *arXiv preprint arXiv:2403.11894*, 2024.

[69] K. Huang, J. Altosaar, and R. Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

[70] Z. Huang, Z. Wang, S. Xia, and P. Liu. Olympicarena medal ranks: Who is the most intelligent ai so far? *arXiv preprint arXiv:2406.16772*, 2024.

[71] R. Irwin, S. Dimitriadis, J. He, and E. J. Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.

[72] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[73] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, and X. Lu. PubMedQA: A dataset for biomedical research question answering. 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)*, pages 2567–2577, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[74] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, S. Hao, B. Moody, B. Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023.

[75] P. Kailas, M. Homilius, R. C. Deo, and C. A. MacRae. Notecontrast: Contrastive

language-diagnostic pretraining for medical text. In *Machine Learning for Health (ML4H)*, pages 201–216. PMLR, 2023.

[76] K. R. Kanakarajan, B. Kundumani, A. Abraham, and M. Sankarasubbu. Biosimcse: Biomedical sentence embeddings using contrastive learning. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 81–86, 2022.

[77] J. Kaplan, S. McCandlish, T. Henighan, T. Brown, B. Chess, R. Child, S. Gray, et al. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[78] A. S. Kasmaee, M. Khodadad, M. A. Saloot, N. Sherck, S. Dokas, H. Mahyar, and S. Samiee. Chemteb: Chemical text embedding benchmark, an overview of embedding models performance & efficiency on a specific domain. *arXiv preprint arXiv:2412.00532*, 2024.

[79] M. Khodadad, A. S. Kasmaee, M. Astaraki, N. Sherck, H. Mahyar, and S. Samiee. Evaluating multi-hop reasoning in large language models: A chemistry-centric case study. *arXiv preprint arXiv:2504.16414*, 2025.

[80] M. Khodadad, A. Shiraee, M. Astaraki, and H. Mahyar. Towards domain specification of embedding models in medicine. *arXiv preprint arXiv:2507.19407*, 2025.

[81] E. Kiciçman, R. Ness, A. Sharma, and C. Tan. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050v3*, 2024.

[82] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker,

P. A. Thiessen, B. Yu, et al. Pubchem in 2021: new data content and improved web interfaces. *Nucleic acids research*, 49(D1):D1388–D1395, 2021.

[83] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1): D1373–D1380, 2023.

[84] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. doi: 10.1088/2632-2153/aba947.

[85] A. Krithara, A. Nentidis, K. Bougiatiotis, and G. Paliouras. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10:170, 2023. doi: 10.1038/s41597-023-02015-7.

[86] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[87] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[88] S. Langer, F. Neuhaus, and A. Nürnberg. CEAR: Automatic construction of a knowledge graph of chemical entities and roles from scientific literature. *arXiv preprint arXiv:2407.21708*, 2024.

[89] A. Lavie and A. Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, 2007.

[90] T. Le Scao and B. Collaboration. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[91] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(Suppl$_1$) : $i103 - -i110, 2020$.

[92] K. Lekadir, M. Prosperi, S. Murphy, and 2025members=100 et al. Future-ai: International consensus guideline for trustworthy and deployable ai in healthcare. *BMJ*, 388:bmj–2024–081554, 2025. doi: 10.1136/bmj-2024-081554.

[93] O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pages 2177–2185, 2014.

[94] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. K"uttler, M. Lewis, W.-t. Yih, T. Rockt"aschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020.

[95] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. K"uttler, M. Lewis, W.-t. Yih, T. Rockt"aschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020.

[96] J. Li, D. Zhang, X. Wang, Z. Hao, J. Lei, Q. Tan, C. Zhou, W. Liu, Y. Yang, X. Xiong, W. Wang, Z. Chen, W. Wang, W. Li, S. Zhang, M. Su, W. Ouyang, Y. Li,

and D. Zhou. Chemvlm: Exploring the power of multimodal large language models in chemistry. *arXiv preprint arXiv:2408.07246*, 2024.

[97] Y. Li, S. Wu, C. Smith, T. Lo, and B. Liu. Improving clinical note generation from complex doctor-patient conversation. *arXiv preprint arXiv:2408.14568*, 2024.

[98] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

[99] Y. Liang, N. Duan, Y. Gong, N. Wu, F. Guo, W. Qi, M. Gong, L. Shou, D. Jiang, G. Cao, X. Fan, R. Zhang, R. Agrawal, E. Cui, S. Wei, T. Bharti, Y. Qiao, J.-H. Chen, W. Wu, S. Liu, F. Yang, D. Campos, R. Majumder, and M. Zhou. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, 2020.

[100] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, 2004.

[101] D. A. Lindberg, B. L. Humphreys, and A. T. McCray. The unified medical language system. *Yearbook of medical informatics*, 2(01):41–51, 1993.

[102] H. Liu, Z. Wang, X. Chen, Z. Li, F. Xiong, Q. Yu, and W. Zhang. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation. *arXiv preprint arXiv:2502.12442*, 2025.

[103] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[104] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[105] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774, 2017.

[106] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu. Biogpt: Generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac293, 2022.

[107] S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, P. Payne, M. Seneviratne, P. Gamble, , and others (Google Research). Large language models encode clinical knowledge. *Nature*, 620:172–180, 2023.

[108] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[109] medRxiv. medrxiv, 2019. URL `https://www.medrxiv.org/`. accessed 28 Jun 2025.

[110] T. Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[111] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT 2013*, pages 746–751, 2013.

[112] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 3111–3119, 2013.

[113] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[114] L. Min, Z. Fan, F. Dou, J. Sun, C. Luo, and Q. Lv. Adaption bert for medical information processing with chatgpt and contrastive learning. *Electronics*, 13(13): 2431, 2024.

[115] J. H. Morris, K. Soman, R. E. Akbas, X. Zhou, B. Smith, E. C. Meng, C. C. Huang, G. Cerono, G. Schenk, A. Rizk-Jackson, et al. The scalable precision medicine open knowledge engine (spoke): a massive knowledge graph of biomedical information. *Bioinformatics*, 39(2):btad080, 2023.

[116] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

[117] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[118] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)*, pages 2014–2037, 2023.

[119] M.-L. M.-F. Multi-Granularity. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

[120] National Library of Medicine (US). Pubmed, 1996. URL `https://pubmed.ncbi.nlm.nih.gov/`. Updated 30 May 2025; accessed 28 Jun 2025.

[121] National Library of Medicine (US). Clinicaltrials.gov, 2000. URL `https://clinicaltrials.gov/`. Updated 18 Jun 2025; accessed 28 Jun 2025.

[122] National Library of Medicine (US). Pubmed central, 2000. URL `https://www.ncbi.nlm.nih.gov/pmc/`. Updated 12 Jun 2025; accessed 28 Jun 2025.

[123] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, 2014.

[124] H. Nori, N. King, S. M. McKinney, D. Carignan, and E. Horvitz. Capabilities of GPT-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*, 2023.

[125] Z. Nussbaum, J. X. Morris, B. Duderstadt, and A. Mulyar. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*, 2024.

[126] M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48, 2017. doi: 10.1186/s13321-017-0235-x.

[127] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[128] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[129] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[130] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 27730–27744, 2022.

[131] A. Pal, L. K. Umapathi, and M. Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR, 2022.

[132] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[133] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[134] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[135] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[136] E. Pereira. Msds-opp: Operator procedures prediction in material safety data sheets. In *15th Doctoral Symposium*, page 42, 2020.

[137] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2227–2237, 2018.

[138] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

[139] D. Polykovskiy and et al. Molecular sets (moses): A benchmarking platform for molecular generation models. In *Frontiers in Pharmacology*, 2020.

[140] D. M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[141] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018. Technical report.

[142] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *OpenAI Technical Report*, 2018.

[143] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. Technical report.

[144] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. Technical report.

[145] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2020.

[146] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[147] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506, 2020.

[148] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):86, 2021.

[149] N. Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[150] F. Remy, K. Demuynck, and T. Demeester. BioLORD–2023: Semantic textual representations fusing large language models and clinical knowledge graph insights. *Journal of the American Medical Informatics Association*, 31(9):1844–1855, 2024. doi: 10.1093/jamia/ocae029.

[151] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

[152] E. Roman, U. Hahn, and N. H. Shah. Mednli: A natural language inference dataset for the clinical domain. *arXiv preprint arXiv:1808.06752*, 2018.

[153] A. Romanov and C. Shivade. Lessons from Natural Language Inference in the Clinical Domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 1586–1596, 2018.

[154] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the Joint Conference on EMNLP-CoNLL*, pages 410–420, 2007.

[155] P. Ruas and F. M. Couto. Nilinker: attention-based approach to nil entity linking. *Journal of Biomedical Informatics*, 132:104137, 2022.

[156] P. Sahoo, P. Meharia, A. Ghosh, S. Saha, V. Jain, and A. Chadha. A comprehensive survey of hallucination in large language, image, video and audio foundation models. In *Findings of EMNLP*, pages 11709–11724, 2024.

[157] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and

T. Lillicrap. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30, 2017.

[158] T. Schick, J. Dwivedi-Yu, R. Dessi, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

[159] P. Schwaller, R. Petraglia, V. Zullo, M. Naderi, and T. Laino. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 2019.

[160] P. Schwaller, R. Petraglia, V. Zullo, M. Naderi, and T. Laino. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 2019.

[161] M. H. S. Segler, T. Kogej, C. Tyrchan, and M. P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4 (1):120–131, 2018. doi: 10.1021/acscentsci.7b00512.

[162] R. Sever, T. Roeder, S. Hindle, and et al. biorxiv: the preprint server for biology. *bioRxiv*, 2019. doi: 10.1101/833400. preprint.

[163] Z. Shi, W. Sun, S. Gao, P. Ren, Z. Chen, and Z. Ren. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. *arXiv preprint arXiv:2406.14891*, 2024.

[164] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record analysis. *IEEE J.*

Biomedical and Health Informatics, 22(5):1589–1604, 2018. doi: 10.1109/JBHI.2017.2767063.

[165] C. Shivade, N. Pourdamghani, F. Nan, P. Resnik, D. Oard, P. Bhatia, et al. Towards clinical encounter summarization: Learning to compose discharge summaries from prior notes. In *Proceedings of NAACL-HLT 2015*, 2015.

[166] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, M. Amin, L. Hou, K. Clark, S. R. Pfohl, H. Cole-Lewis, D. Neal, Q. M. Rashid, M. Schaekermann, A. Wang, D. Dash, J. H. Chen, and N. H. Shah. Toward expert-level medical question answering with large language models. *Nature Medicine*, 30:1134–1142, 2024.

[167] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, M. Amin, L. Hou, K. Clark, S. R. Pfohl, H. Cole-Lewis, D. Neal, Q. M. Rashid, M. Schaekermann, A. Wang, D. Dash, J. H. Chen, and N. H. Shah. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31:943–950, 2025.

[168] G. Soganciogl u, H. Öztürk, and A. Özgür. BIOSSES: A semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58, 2017.

[169] M. Sorokina, P. Merseburger, K. Rajan, M. A. Yirik, and C. Steinbeck. Coconut online: collection of open natural products database. *Journal of Cheminformatics*, 13(1):2, 2021.

[170] I. Sukeda, M. Suzuki, H. Sakaji, and S. Kodera. Development and analysis of medical instruction-tuning for Japanese large language models. *Artificial Intelligence in Health*, 1(2):107–116, 2024. doi: 10.36922/aih.2695.

[171] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, 2014.

[172] W. Tai, H. Kung, X. L. Dong, M. Comiter, and C.-F. Kuo. exbert: Extending pre-trained models with domain-specific vocabulary under constrained training resources. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1433–1439, 2020.

[173] X. Tang, D. Shao, J. Sohn, J. Chen, J. Zhang, J. Xiang, F. Wu, Y. Zhao, C. Wu, W. Shi, A. Cohan, and M. Gerstein. Medagentsbench: Benchmarking thinking models and agent frameworks for complex medical reasoning. *arXiv preprint arXiv:2503.07459*, 2025.

[174] Y. Tang and Y. Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*, 2024.

[175] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[176] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Datasets and Benchmarks Track*, 2021.

[177] S. T. I. Tonmoy, S. M. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*, 2024.

[178] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozi‘ere, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[179] P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

[180] D. van Veen, C. Van Uden, L. Blankemeier, J.-B. Delbrouck, A. Aali, C. Bluethgen, A. Pareek, M. Polacin, R. E. Pontes, A. Seehofnerova, N. Rohatgi, P. Hosamani, W. Collins, N. Ahuja, C. P. Langlotz, J. Hom, S. Gatidis, J. Pauly, and A. S. Chaudhari. Adapted large language models can outperform medical experts in clinical text summarization. *Nature Medicine*, 30:1134–1142, 2024.

[181] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5998–6008, 2017.

[182] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[183] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[184] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A

multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop on BlackboxNLP*, pages 353–355, 2018.

[185] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[186] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2019.

[187] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2020.

[188] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

[189] R. Wang, R.-X. Wang, M. Manjrekar, and C. W. Coley. Neural graph matching improves retrieval-augmented generation in molecular machine learning. *arXiv preprint arXiv:2502.17874*, 2025.

[190] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[191] Y. Wang, N. Afzal, S. Fu, F. Shen, M. Rastegar-Mojarad, and H. Liu. MedSTS: A resource for clinical semantic textual similarity. *Language Resources and Evaluation*, 54:57–72, 2020.

[192] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[193] J. Welbl, P. Stenetorp, and S. Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018.

[194] G. Wellawatte, H. Guo, M. Lederbauer, A. Borisova, M. Hart, M. Brucka, and P. Schwaller. Chemlit-qa: A human evaluated dataset for chemistry rag tasks. In *AI for Accelerated Materials Design-NeurIPS 2024*.

[195] G. P. Wellawatte, H. Guo, M. Lederbauer, A. Borisova, M. Hart, M. Brucka, and P. Schwaller. Chemlit-qa: a human evaluated dataset for chemistry rag tasks. *Machine Learning: Science and Technology*, 6(2):020601, 2025.

[196] Wikipedia contributors. Wikipedia, The Free Encyclopedia. `https://www.wikipedia.org/`, 2025. [Online; accessed 22 July 2025].

[197] A. Williams, N. Nangia, and S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2018)*, pages 1112–1122, 2018.

[198] M. Wu, Y. Wang, Y. Ming, Y. An, Y. Wan, W. Chen, B. Lin, Y. Li, T. Xie, and D. Zhou. Chemagent: Enhancing llms for chemistry and materials science through tree-search based tool learning. *arXiv preprint arXiv:2506.07551*, 2025.

[199] X. Wu, Y. Zhang, J. Yu, W. Zhao, Y. Guo, C. Zhang, H. Liu, J. Shang, X. Deng, and Y. Gong. Virtual data augmentation method for reaction prediction. *Scientific Reports*, 12:17098, 2022. doi: 10.1038/s41598-022-21524-6.

[200] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, K. Leswing, and V. Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical Science*, 9:513–530, 2018.

[201] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649, 2024.

[202] G. Xiong, Q. Jin, Z. Lu, and A. Zhang. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6233–6251, 2024. doi: 10.18653/v1/2024.findings-acl.372.

[203] L. Xu, H. Hu, X. Zhang, L. Li, C. Cao, Y. Li, Y. Xu, K. Sun, D. Yu, C. Yu, Y. Tian, Q. Dong, W. Liu, B. Shi, Y. Cui, J. Li, J. Zeng, R. Wang, W. Xie, Y. Li, Y. Patterson, Z. Tian, Y. Zhang, H. Zhou, S. Liu, Z. Zhao, Q. Zhao, C. Yue, X. Zhang, Z. Yang, K. Richardson, and Z. Lan. CLUE: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*, pages 4762–4772, 2020.

[204] L. Xu, S. Pan, L. Xia, and Z. Li. Molecular property prediction by combining LSTM and GAT. *Biomolecules*, 13(3):503, 2023. doi: 10.3390/biom13030503.

[205] X. Yang, A. Chen, N. PourNejatian, H. C. Shin, K. E. Smith, et al. Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records. *arXiv preprint*, arXiv:2203.03540, 2022. URL `https://arxiv.org/abs/2203.03540`.

[206] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2369–2380, 2018.

[207] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[208] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 5754–5764, 2019.

[209] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[210] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[211] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

[212] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Y. Chen, , et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[213] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019.

[214] X. Zhang, M. Wang, X. Yang, D. Wang, S. Feng, and Y. Zhang. Hierarchical retrieval-augmented generation model with rethink for multi-hop question answering. *arXiv preprint arXiv:2408.11875*, 2024.

[215] Y. Zhang, J. Baldridge, and L. He. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019)*, pages 1298–1308, 2019.

[216] Y. Zhang, Z. Shen, C.-H. Wu, B. Xie, J. Hao, Y.-Y. Wang, K. Wang, and J. Han. Metadata-induced contrastive learning for zero-shot multi-label text classification. In *Proceedings of the ACM Web Conference 2022*, pages 3162–3173, 2022.

[217] X. Zhong, B. Jin, S. Ouyang, Y. Shen, Q. Jin, Y. Fang, Z. Lu, and J. Han. Benchmarking retrieval-augmented generation for chemistry. *arXiv preprint arXiv:2505.07671*, 2025.