Benchmarking and Analyzing In-Context Learning, Fine-tuning and Supervised Learning for Biomedical Knowledge Curation: a focused study on chemical entities of biological interest

Emily Groves*
Institute of Health Informatics, University College London
London, United Kingdom
emily.groves.22@ucl.ac.uk

Minhong Wang*
Institute of Health Informatics, University College London
London, United Kingdom
minhong.wang@ucl.ac.uk

Yusuf Abdulle
Institute of Health Informatics, University College London
London, United Kingdom
yusuf.abdulle.20@ucl.ac.uk

Holger Kunz
Institute of Health Informatics, University College London
London, United Kingdom
h.kunz@ucl.ac.uk

Jason Hoelscher-Obermaier
iris.ai
Oslo, Norway
jason@iris.ai

Ronin Wu
iris.ai
Oslo, Norway
ronin@iris.ai

Honghan Wu
Institute of Health Informatics, University College London
London, United Kingdom
honghan.wu@ucl.ac.uk

ABSTRACT

Automated knowledge curation for biomedical ontologies is key to ensure that they remain comprehensive, high-quality and up-to-date. In the era of foundational language models, this study aims to compare and analyze three natural language processing (NLP) paradigms for curation tasks: in-context learning (ICL), fine-tuning (FT) and supervised learning (ML). Chemical Entities of Biological Interest (ChEBI) database was used as an exemplar ontology, on which three curation tasks were devised. GPT-4, GPT-3.5 and BioGPT were utilized for ICL using three prompting strategies. PubmedBERT was chosen for the FT paradigm. For ML, six embedding models were utilized for training Random Forest and Long-Short Term Memory models. To assess different paradigms' utilities in different data availability scenarios, five different setups were configured for assessing the effect on ML and FT performances. The full datasets generated for curation tasks were task 1 (#Triples 620,386), task 2 (611,430) and task 3 (617,381), with a 50:50 positive versus negative ratio. For ICL models, GPT-4 achieved best accuracy scores of 0.916, 0.766 and 0.874 for tasks 1-3 respectively. In a head-on-head comparison, ML (trained on around 260,000 triples) was more accurate than ICL in all tasks (accuracy differences: +.11, +.22 and +.17). Fine-tuned PubmedBERT performed similarly to best ML models in tasks 1 & 2 (F1 differences: -.014 and +.002), but worse in task 3 (-.048). Simulation experiments showed both ML and FT models deteriorated in smaller and higher-imbalanced training data. When training data had 6,000 triples or fewer, GPT-4 was superior to ML/FT models in tasks 1 and 3. However, ICL never performed on par with the ML/FT in task 2. When prompted properly, foundation models with ICL can be good assistants for knowledge curation, however, clearly not yet to a level making ML and FT paradigms obsolete. The latter two need good task-related training data to outperform ICL. Notably, in such situations, the ML paradigm only needs small pretrained embedding models and much less computation.

VLDB Workshop Reference Format:
Emily Groves*, Minhong Wang*, Yusuf Abdulle, Holger Kunz, Jason Hoelscher-Obermaier, Ronin Wu, and Honghan Wu. Benchmarking and Analyzing In-Context Learning, Fine-tuning and Supervised Learning for Biomedical Knowledge Curation: a focused study on chemical entities of biological interest. VLDB 2024 Workshop: LLM+KG.

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/knowlab/AutomatedKGEnrichment/.

1 INTRODUCTION

Knowledge Graphs (KGs) [11, 24] are a novel paradigm for integrating and representing semantically networked datasets or knowledge bases from highly heterogenous sources. They are well-suited to the large and heterogeneous datasets common in the biomedical domain [19]. Unsurprisingly, there is a large body of literature of utilising KGs for biomedical purposes including automated diagnosis [15, 34], generation of radiology reports [38], and pharmaceutical studies [1, 16, 36].

Proceedings of the VLDB Endowment. ISSN 2150-8097.

KGs can suffer from sparsity and incompleteness [5], and also require updating periodically as new information or knowledge becomes available. Manual KG curation is, however, time-consuming, burdensome and impractical in settings where the pace of knowledge generation is high. Automated knowledge graph enrichment or refinement [25] is a subfield in graph-based machine learning, focused on correctly integrating new entities into existing knowledge graphs [29] and predicting novel relationships between entities. This can significantly enhance the efficiency of the curation processs [13].

New knowledge and information are typically first presented in free-text format, e.g., scientific literature [27], news articles and social media [21]. Due to the nature of these sources, it is not surprising to see Natural Language Processing (NLP) techniques [14] play an instrumental role in the creation and curation of KGs by automating effective information extraction at scale [9, 33, 35].

Powerful, transformer-based Large Language Models (LLMs) have emerged in recent years [31, 32], significantly transforming Natural Language Processing. Via task-agnostic, self-supervised pre-training on vast corpora, these models learn lexical, syntactic, and semantic structures. In particular, the emergence of foundation models like GPT3.5/GPT4.0 and the open source LLMs (e.g., Llama2 [30], Mistral [12]) has had an undeniable impact on NLP research. They are speeding up, or at least stimulating discussion on, paradigm shifts from supervised learning to fine-tuning to prompting/in-context learning [17], and from development of models specialised for a single function to versatile, general-purpose models which can be applied to a wide array of tasks. In this state of flux, for automated biomedical KG enrichment, it is sensible to ask:

- How do these foundational LLMs perform in curating biomedical knowledge, including differences between models and effectiveness of various in-context learning strategies?
- Can smaller, domain-specific language models compete with large, open domain state-of-the-art LLMs?
- Are supervised learning approaches truly obsolete in such tasks?

This study aims to conduct a series of experiments for answering these questions. The Chemical Entities of Biological Interest (ChEBI) database will be used for this focused study. ChEBI is one example of knowledge graphs in the interdisciplinary field between chemistry and biomedicine. It functions as both a database and an ontology, housing information about chemical entities of biological relevance, and contains a diverse array of curated data items [6, 7]. This resource finds extensive application across various domains, including drug target identification [18, 28] and gene studies [2]. However, it is worth noting that in the case of ChEBI, the addition of new entities and connections is a manual process [23], which translates to a substantial investment of time and resources.

2 MATERIALS AND METHODS

A schematic representation of this work is provided in Figure 1. The core (the box on the right of Figure 1) is a set of experiments to evaluate three paradigms of applying NLP in automated knowledge curation for the ChEBI KG: (1) in-context learning with pretrained large language models; (2) fine-tuning a pretrained BERT (Bidirectional Encoder Representations from Transformers [8]) model with task-related training data; (3) supervised machine learning approaches using distributed representations. A diverse set of models (depicted in the two-dimensional space on the left) was utilized in this study, plotted in the space indicating their training corpus size (x-axis) and domain relevance (y-axis). Models which underwent further training on task related corpora are indicated in yellow.

Three types of enrichment tasks were proposed to assess the models' abilities in detecting different forms of 'erroneous' knowledge (the top dashed-line box on the right of Figure 1). We also simulated five scenarios where the size and imbalance (negative vs positive data labels) of the training data vary (the bottom dashed-line box in Figure 1). This is to assess how fine-tuning and supervised learning approaches perform in different situations, which will in turn give evidence on how to choose NLP paradigms, e.g., the settings where foundation models may be most useful.

2.1 ChEBI database

The publicly available ChEBI knowledge graph[1] was downloaded in February 2022 for comparing different approaches in knowledge graph enrichment tasks. We included data from all three sub-ontologies and nine of the ten ChEBI relationship types. For simplicity, the relationship 'is conjugate acid of', which is the inverse relationship of 'is conjugate base', was removed.

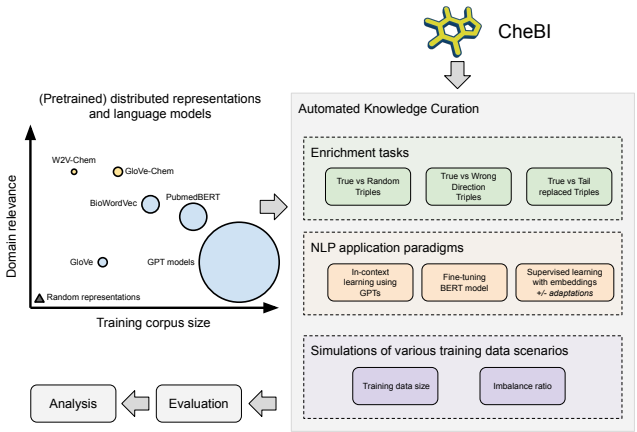


Figure 1: Architecture of study design. Pretrained distributed representations and language models panel (top-left): X-axis is the size of corpus for pretraining the models. Y-axis is the relevance of the model pretraining corpus to the target domain. The size of the shapes denotes the model size. (Note: ratios are indicative only.) Three simulated knowledge curation tasks were devised, and different model adaptation techniques proposed. Different difficulty settings were adjusted via modification of training data size and imbalance ratio.

[1] https://www.ebi.ac.uk/chebi/downloadsForward.do

2.2 Knowledge enrichment tasks

The ChEBI ontology is denoted as $G = (V, T, L)$, where:

- V is a set of nodes, representing all entities in ChEBI;
- T is a set of triples, where each triple $t = (s, o, l)$ consists of two nodes s and o, called the subject and object of the triple, and a label l;
- L is a set of labels, representing the possible relationships in ChEBI.

The enrichment task defined in this work is a simple binary classification

$$f(t) = \begin{cases} 1, & \text{if } t \text{ is a correct triple} \\ 0, & \text{otherwise} \end{cases}$$

A triple is called *correct* if it denotes a true piece of knowledge.

Three binary classification tasks were devised in simulated knowledge enrichment tasks of identifying different types of erroneous triples:

- **Task one (true vs random false triples)**: Positive triples comprised those extracted from the ChEBI database, i.e., $T_{pos} \subseteq T$. Negative triples were randomly generated as those without a directed link from a subject entity to an object entity, i.e., $T_{neg1} \subseteq \{(s, o, l) | s \in V, o \in V, (s, o, l) \notin T\}$.
- **Task two (true vs wrong direction triples)**: This task assessed the degree to which models could distinguish true triples (T_{pos}) from flipped negative triples. Negative triples were generated by inversion of positive triples, $T_{neg2} \subseteq \{(o, s, l) | (s, o, l) \in T, (o, s, l) \notin T\}$. For example, for a positive triple *(Androsta-4,9(11)-diene-3,17-dione, has_role, androgen)*, the corresponding negative triple would be *(androgen, has_role, Androsta-4,9(11)-diene-3,17-dione)*.
- **Task three (true vs wrong object triples)** In the final and probably also most challenging task, models were asked to differentiate between true triples and their counterparts where only the object was replaced with a closely related entity, i.e., one of its sibling entities in ChEBI: $T_{neg3} \subseteq \{(s, o_2, l) | (s, o_1, l) \in T, (s, o_2, l) \notin T, p(o_1) \cap p(o_2) \neq \emptyset\}$, where $p(\cdot)$ is the function to retrieve parents of an entity. For example, for a positive triple *(Androsta-4,9(11)-diene-3,17-dione, has_role, androgen)*, a negative triple could be *(Androsta-4,9(11)-diene-3,17-dione, has_role, estrogen)*.

2.3 Pretrained language models and distributed representations

The following pretrained causal language models were included in this study for in-context learning experiments:

- BioGPT [20]: a domain-specific generative Transformer language model pretrained on large-scale biomedical literature, comprising 15 million PubMed items [2], each with both title and abstract, retrieved before 2021.
- OpenAI's GPT model versions 3.5 and 4.0. Both models were accessed via OpenAI's API access point[3]. The GPT3.5 model used was gpt-3.5-turbo with model ID of *gpt-3.5-turbo-0613*[4] and the utilized GPT4.0 model name was gpt4 with an ID of *gpt-4-0613*[5].

We used the PubmedBERT [10] model in language model fine-tuning experiments. For the third NLP paradigm experiments, supervised learning, the following (pretrained) embedding models were included.

- **GloVe [26]:** GloVE (Global Vectors for Word Representation) is an embedding model pretrained using an unsupervised learning algorithm on the Common Crawl corpus with a total of 840B tokens. This study used the *glove.840B.300d* obtained from https://nlp.stanford.edu/projects/glove/, which has a 2.2M cased vocabulary and a vector dimension size of 300.
- **W2V-Chem:** A word2vec [22] model was trained from scratch on 7,201 full papers from the chemical domain. These were sourced from PubMed using cross references associated with the ChEBI ontology. The titles, abstracts and full texts of these papers were used to train domain-specific word embeddings. Embeddings were initialized from random vectors.
- **GloVe-Chem:** We developed an embedding model by further adapting the GloVe embeddings for ChEBI enrichment. Specifically, the previously mentioned 7,201 PubMed articles were used to further train embeddings from the GloVe model. In contrast to W2V-Chem, the vocabulary was built by joining the texts from chemical domain papers and the vocabulary from GloVe. The input layer was initialised from Glove embeddings.
- **Biowordvec [37]:** An embedding model was developed for the biomedical domain using fastText [4] trained on large biomedical corpus, as well as on information from the MeSH knowledge graph.
- **PubmedBERT embeddings**: PubmedBERT was also used for deriving vector representations for triples. We summed up the last 4 hidden layers of the special token [CLS] for each component of a triple and used this as the entity representation.
- **Random embeddings:** We were also interested in evaluating word representations with no semantics by using a random embedding model. For a given ChEBI database entity, embeddings were generated via tokenization followed by assignment of a 300-dimension vector to each token. Vectors were randomly generated from uniform distribution between -1 and 1.

2.4 NLP paradigm 1: In-context learning with pretrained Large Language Models

Relative performances of GPT-3.5 Turbo, GPT-4 and BioGPT in the three binary classification tasks were evaluated using few-shot prompting. Models were provided with three positive and three negative example triples, and prompted to classify a seventh. Triples used in prompts were selected randomly from training data, eliminating any duplicates. Approximately equal numbers of positive

[2] https://pubmed.ncbi.nlm.nih.gov
[3] https://api.openai.com/v1/chat/completions
[4] https://platform.openai.com/docs/models/gpt-3-5
[5] https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo

and negative triples were queried for classification. We selected a 50:50 ratio to ensure that there was balanced training in this work

For each task, models were provided with 100 distinct prompts, with each repeated five times. We experimented with three prompt formulations: (i) a base prompt (Table 1); (ii) a second variant in which we added an additional sentence *'If you do not know the answer, state 'I don't know'*, aiming to reduce hallucinations, and hence improve the utilities and performances; and (iii) a variant in which positive and negative example triples were presented in a random order. The latter was done in response to an observed tendency for the BioGPT model to disproportionately classify triples as negative, having hypothesized that this might result from the order in which triples were presented (three positive examples, three negative examples, query triple).

2.5 NLP paradigm 2: fine-tuning BERT model for knowledge curation

The PubmedBERT model was fine-tuned to conduct each of the three classification tasks. Triples were converted into sequences of words by concatenating the labels of subject, relationship and object with a special separator token <*SEP*>. The sequence was then tokenized using the PubmedBERT tokenizer, and the output was fed into the transformer layers. The final layer of the model is a fine-tuning layer specific to the document classification task. This layer takes the output of the encoder layer and applies a feed-forward neural network to produce a vector representation of the document. This vector representation is then passed through a softmax layer to produce a probability distribution over the possible document classes (true or false in our scenario).

For supervised learning approaches, we follow the same process as formalized in Algorithm 1 for using different embedding models and different learning algorithms. Essentially, triples were converted to vector representations, which were then fed into the chosen machine learning (ML) algorithm for model fitting. The vector representations were generated depending on the chosen algorithm type:

- If the ML algorithm is a recurrent neural network (RNN) or its variants (e.g., LSTM - Long Short Term Memory network), the representation of a triple will be a sequence of vectors generated by (1) tokenizing each component of the triple; (2) converting each token into a vector using the chosen embedding model; (3) merging three vector sequences by using a special separator token, indicating the boundaries of the components.
- For other algorithms, triples will be converted into one vector by (1) tokenizing each component of the triple; (2) averaging vectors of each component; (3) combining the representations of each of the three components by concatenation.

LSTM and Random Forest ML algorithms were chosen in our implementations, representing RNN and non-sequential archetypes. Two types of tokenizers were used; for the PubmedBERT embedding model, the PubmedBERT tokenizer was used. For all other embedding models, we used the the NLTK [3] library. Specifically, its RegexpTokenizer [6] was used with hand-crafted regular expression patterns for tokenizing special chemical entity names. Random vectors were used for out of vocabulary situations. Hyperparameter optimization was applied using a 5-fold cross validation on training data, optimized for F1-scores.

Algorithm 1 Supervised learning for knowledge curation using embeddings

Input:
X: the training data - a list of triples;
y: the labels of the training data;
emb: an embedding model;
sep_token: a special token as separator;
Tkn: a tokenizer;
M: the supervised learning algorithm;
$model\_type$: the type of the M.
Output: Fitted model m.

$X_v \leftarrow []$
$v_{sep} \leftarrow Emb(sep\_token)$
for t in X **do**
  $(s, o, l) \leftarrow t$
  if $model\_type$ is RNN **then** ▷ For RNN-like algorithms, generate a sequence of vectors
    $(w_{s1}, ..., w_{si}) \leftarrow Tkn(s)$
    $(w_{l1}, ..., w_{lj}) \leftarrow Tkn(l)$
    $(w_{o1}, ..., w_{om}) \leftarrow Tkn(o)$
    $vect \leftarrow [$
      $emb(w_{s1}), ..., emb(w_{sj}), v_{sep},$
      $emb(w_{l1}), ..., emb(w_{lj}), v_{sep},$
      $emb(w_{o1}), ..., emb(w_{oj})$
    $]$
  else
    $vect \leftarrow concatenate($
      $\frac{\sum_{w \in Tkn(s)} emb(w)}{|Tkn(s)|},$
      $\frac{\sum_{w \in Tkn(l)} emb(w)}{|Tkn(l)|},$
      $\frac{\sum_{w \in Tkn(o)} emb(w)}{|Tkn(o)|})$
  end if
  $X_v.append(vect)$
end for
$m \leftarrow M.fit(X_v, y)$
return m

2.6 Effects of imbalanced data and variations in training data size

It is well understood that ML model performances are adversely affected by training data imbalance and/or scarcity. We therefore sought to explore and compare model performances (paradigms 2 & 3: fine-tuning and supervised learning) under these sub-optimal conditions. Such an investigation would also provide evidence on when the paradigm 1 is most useful, since pretrained LLMs have been trained on large corpora and thus have minimal dependency on task-related training data.

[6] https://www.nltk.org/api/nltk.tokenize.RegexpTokenizer.html

Basic prompt template

```
"""
Your task is to classify triples as True or False.
<triple>: {positive_example_1}
<classification>: True
<triple>: {positive_example_2}
<classification>: True
<triple>: {positive_example_3}
<classification>: True
<triple>: {negative_example_1}
<classification>: False
<triple>: {negative_example_2}
<classification>: False
<triple>: {negative_example_3}
<classification>: False
<triple>: {prompt_triple}
"""
```

Table 1: The template used for prompting LLMs (Variant #1). In generating prompts, curly bracket-enclosed contents were replaced with corresponding triples derived from actual data.

Using reduced datasets (~10% of the full training and test datasets), we generated varying train-test split ratios via random selection of successively smaller subsets of training data (9:1, 8:1, 7.1, 6:1, 5:1, 4:1, 3:1, 2:1, 1:1, 0.5:1 for training:testing splits). Effects of imbalanced data were determined by altering the ratio of positive versus negative triples present in training data (1:1, 0.75:1, 0.5:1, 0.25:1, 0.125:1 positive:negative).

3 RESULTS

As of February 2022, ChEBI contained 147,461 entities. Chemical Entities represent the majority (145,869), followed by 1,550 Role Entities and 42 Subatomic Particles. There are total 318,438 triples, with a highly imbalanced distribution of relationships; the 3 most common types make up > 90% of all triples: 230,241 (72.3%) *is_a*, 42,095(13.2%) *has_role* and 18,204 (5.7%) *has_functional_parent*. A total of 47,701 unique tokens were derived from these triples using the NLTK tokenization process described in the method section. Table 2 shows the detailed numbers of the populated datasets for three tasks and setups for three NLP paradigms.

3.1 Results of supervised learning paradigm

Table 3 shows the results of random forest models on all three tasks. For task 1, W2V-Chem, a word2vect model trained from scratch, performed best (F1-score: 0.9690). Performances of LSTM models in general were on par with those of random forest models. For the simplicity of reporting and discussions, LSTM models' result were not presented or discussed in the rest of this paper. The best performing embedding model for task 2 was PubmedBERT, while GloVe-Chem (GloVe futher trained on ChEBI related papers) was the best for task 3.

Comparing the best F1-scores across three tasks (Task 1: 0.9690, Task 2: 0.9822 and Task 3: 0.9125), it seemed Task 3 (in which negative triples were formed by replacing the object with similar entities) was the most challenging for ML based approaches and Task 2 (in which negative triples were formed by swapping the relationship direction) was the easiest.

3.2 Results of fine-tuning PubmedBERT

We fine-tuned the PubmedBERT model for three document classification tasks, and utilised a Cross-Entropy loss function. This model ran for 3 epochs and found that there was negligible differences in performance when running between 3 and 4 epochs, and thus optimised for performance. The learning rate for this model was set to 1×10^{-4} and employed the Adam optimiser. The fine-tuning datasets for three tasks and results of fine-tuned PubmedBERT are summarised in Table 4. Overall, performances are on par with Random Forest model trained on PubmedBERT embeddings, and frequently rank among the best approaches tested, although not consistently so. In particular, the fine-tuned PubmedBERT model for task 3 was about 5% worse than the best ML based model (Random forest using GloVe-Chem).

3.3 Results of In-context learning paradigm: prompting three GPT models

Table 5 contains results for LLM prompting experiments for classification of true versus randomly-generated negative triples (Task 1), true versus reversed triples (Task 2) or true versus closely-related negative triples (Task 3). The column *No. unclassified* shows the numbers of triples for which the model either did not give a valid result (True of False) or explicitly said 'I don't know' in our second prompting strategy. These triples were deemed as not accurately classified in *accuracy* evaluation. However, they were excluded in *precision*, *recall* and *F1* calculations. This was done in order to comprehensive evaluate both LLMs' general performances on all tests, and performances where a decisive answer was given.

Table 2: Statistics of generated datasets for three tasks. Training and test sets shown are for the supervised learning paradigm, which was based on a split of 9:1 ratio.

	Triples		Training set		Test set		Total
	#positive	#negative	#positive	#negative	#positive	#negative	
Task 1	310,193	310,193	279,178	279,177	31,015	31,016	620,386
Task 2	305,715	305,715	275,146	275,146	30,569	30,569	611,430
Task 3	310,193	307,188	279,178	276,469	31,015	30,719	617,381

Table 3: Results of NLP Paradigm 1: supervised machine learning using embedding models. These are results from Random Forest models. Bold texts indicate the best performances.

	Task 1			Task 2			Task 3		
Embeddings	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Random	0.9576	0.9573	0.9574	0.9581	0.9581	0.9581	0.9042	0.9042	0.9042
GloVe	0.954	0.9536	0.9538	0.9573	0.9573	0.9573	0.9073	0.9073	0.9073
W2V-Chem	**0.9691**	**0.969**	**0.9690**	0.9596	0.9596	0.9596	0.9122	0.9122	0.9122
GloVe-Chem	0.9683	0.9683	0.9683	0.9586	0.9586	0.9586	**0.9126**	**0.9125**	**0.9125**
BioWordVec	0.9676	0.9675	0.9675	0.9605	0.9605	0.9605	0.9062	0.9061	0.9061
PubmedBERT	0.9356	0.9353	0.9354	**0.9822**	**0.9822**	**0.9822**	0.906	0.906	0.9060

Table 4: Results of NLP paradigm 2: Fine-tuning datasets and performances of fine-tuned PubmedBERT models on three tasks

Tasks	Datasets (# Triples)			Model Performance			
	Training	Validation	Test	Accuracy	Precision	Recall	F1
Task 1	496,308	62,039	62,039	0.9565	0.9798	0.9319	0.9552
Task 2	489,144	61,143	61,143	0.9840	0.9931	0.9749	0.9839
Task 3	493,903	61,739	61,739	0.8723	0.9240	0.8124	0.8646

Overall, GPT-3.5 Turbo and GPT-4 achieved competitive performances via few-shot prompting approaches. GPT-4 outperformed GPT-3.5 Turbo by a considerable margin in all three tasks, attaining maximal F1 scores of 0.9041, 0.8880 and 0.9082, respectively. Importantly, both models provided extremely consistent responses, with minimum Fleiss kappa scores of 0.95 and 0.86, respectively.

Performance of BioGPT was comparatively poor, however, with accuracy and Fleiss' kappa scores consistent with random guessing. Modification of the base prompt to allow the models to answer 'I don't know' did not appear to reliably enhance precision and F1 scores, but did generally lead to an increase in proportion of unclassified triples and consequent reduction in overall accuracy. Randomization of the ordering of positive and negative example triples appeared to be a more effective modification. In particular, GPT-4 prompted using this formulation yielded the highest F1 scores in all tasks.

3.4 Comparisons of three NLP paradigms on enrichment tasks

Comparing results from Tables 3-5, across three enrichment tasks, supervised learning approaches and fine-tuning pretrained BERT models achieved similar performances, with the exception of task 3, where the fine-tuned PubmedBERT model performed worse. Both were superior to in-context learning of LLMs, even only considering those triples for which LLMs gave confident answers. However, these might not direct comparable because they were not assessed using the same test set.

3.4.1 Effects of imbalanced data and variations in train-test splits on supervised learning and fine-tuning. Figure 2 shows the changing patterns of F1-scores of three representative models in all three tasks. For each task, we picked three models; models trained using random vectors (as a reference) and two most consistently performing models. Unsurprising, in all tasks, performances decreased steadily as less training data was available and greater imbalance was introduced. PubmedBERT and GloVe-Chem were the most consistent models, i.e., less prone to sub-optimal training data. Fine-tuned models outperformed all ML based approaches in the first two tasks. However, the fine-tuned approach suffered much more significantly for task 3, performing worse even than ML with random embeddings.

We also plotted GPT-4's performances on the figures. Essentially, GPT-4 would be a better tool to use in scenarios where those solid lines are below the dashed line, i.e., ML based or fine-tuning approaches won't achieve any better performances than GPT-4. For task 1, GPT-4 outperformed both ML-based and fine-tuned approaches in the two most extreme scenarios, i.e., (Split: 1:1; P:N: 1:8) and (Split: 0.5:1; P:N: 1:10). For task 3, GPT-4 was superior in all

Table 5: Results of NLP paradigm 3: comparisons of effectiveness and consistency of in-context learning using LLMs for all three tasks with different prompting strategies. Note: for accuracy evaluation, the unclassified triples were included; for other metrics, those were NOT included.

Model	Prompt formulation	Overall accuracy: Mean (SD)	No. unclassified (%)	Precision: Mean (SD)	Recall: Mean (SD)	F1: Mean (SD)	Kappa
GPT-3.5	#1	0.8040 (0.0083)	0 (0)	0.9724 (0.0007)	0.6518 (0.0155)	0.7804 (0.0114)	1.00
	#2	0.7020 (0.0084)	109 (21.8)	1.0000 (0.0000)	0.8067 (0.0025)	0.8930 (0.0015)	0.98
	#3	0.7380 (0.0045)	95 (19.0)	0.9273 (0.0135)	0.8485 (0.0000)	0.8861 (0.0062)	0.97
BioGPT	#1	0.4600 (0.0255)	92 (18.4)	0.4667 (0.3613)	0.0412 (0.0344)	0.0730 (0.0580)	0.07
	#2	0.3500 (0.0224)	111 (22.2)	0.6333 (0.4150)	0.0276 (0.0196)	0.0526 (0.0369)	0.05
	#3	0.4620 (0.0356)	111 (22.2)	0.6530 (0.0892)	0.2872 (0.0671)	0.3979 (0.0814)	0.13
GPT-4	#1	0.9160 (0.0055)	0 (0)	1.0000 (0.0000)	0.8250 (0.0114)	0.9041 (0.0068)	0.98
	#2	0.8660 (0.0152)	27 (5.4)	0.9723 (0.0010)	0.8340 (0.0183)	0.8978 (0.0110)	0.95
	#3	**0.8320 (0.0164)**	**55 (11.0)**	**1.0000 (0.0000)**	**0.8385 (0.0327)**	**0.9119 (0.0195)**	**0.96**

(a) Task 1 - Classification of true versus randomly generated negative triples. Relationship type: 'Is_a'.

Model	Prompt formulation	Overall accuracy: Mean (SD)	No. unclassified (%)	Precision: Mean (SD)	Recall: Mean (SD)	F1: Mean (SD)	Kappa
GPT-3.5	#1	0.6740 (0.0055)	0 (0)	0.7480 (0.0076)	0.6456 (0.0078)	0.6930 (0.0052)	0.97
	#2	0.5920 (0.0045)	97 (19.4)	0.7417 (0.0021)	0.8446 (0.0104)	0.7898 (0.0053)	0.98
	#3	0.5680 (0.0084)	80 (16.0)	0.6264 (0.0080)	0.8342 (0.0109)	0.7155 (0.0085)	0.98
BioGPT	#1	0.3040 (0.0089)	123 (24.6)	0.6667 (0.3118)	0.0349 (0.0186)	0.0656 (0.0345)	0.06
	#2	0.4120 (0.0311)	127 (25.4)	0.4000 (0.2937)	0.0552 (0.0408)	0.0968 (0.0711)	0.08
	#3	0.4180 (0.0415)	103 (20.6)	0.5877 (0.1161)	0.2144 (0.0650)	0.3111 (0.0789)	0.04
GPT-4	#1	0.7660 (0.0134)	0 (0)	0.7650 (0.0150)	0.7680 (0.0110)	0.7665 (0.0127)	0.92
	#2	0.6880 (0.0110)	43 (8.6)	0.7390 (0.0191)	0.7753 (0.0478)	0.7557 (0.0182)	0.86
	#3	**0.8160 (0.0114)**	**38 (7.6)**	**0.8883 (0.0160)**	**0.8880 (0.0182)**	**0.8880 (0.0108)**	**0.94**

(b) Task 2 - Classification of true versus reversed triples. Relationship type: 'Is_a'.

Model	Prompt formulation	Overall accuracy: Mean (SD)	No. unclassified (%)	Precision: Mean (SD)	Recall: Mean (SD)	F1: Mean (SD)	Kappa
GPT-3.5	#1	0.7180 (0.0084)	0 (0)	0.7258 (0.0128)	0.5773 (0.0124)	0.6430 (0.0110)	0.97
	#2	0.6680 (0.0045)	91 (18.2)	0.7838 (0.0000)	0.8056 (0.0000)	0.7945 (0.0000)	0.99
	#3	0.5920 (0.0110)	157 (31.4)	0.8253 (0.0089)	0.9393 (0.0114)	0.8786 (0.0062)	0.95
BioGPT	#1	0.4500 (0.0520)	89 (17.8)	0.4271 (0.1920)	0.0664 (0.0340)	0.1147 (0.0576)	0.01
	#2	0.3440 (0.0207)	88 (17.6)	0.5833 (0.2041)	0.0614 (0.0485)	0.1090 (0.0827)	0.03
	#3	0.4520 (0.0319)	96 (19.2)	0.6854 (0.1036)	0.2989 (0.0499)	0.4152 (0.0650)	0.10
GPT-4	#1	0.8740 (0.0055)	0 (0)	0.9236 (0.0093)	0.8042 (0.0114)	0.8597 (0.0066)	0.94
	#2	0.7980 (0.0045)	54 (10.8)	0.9268 (0.0132)	0.7943 (0.0128)	0.8554 (0.0086)	0.99
	#3	**0.8480 (0.0084)**	**24 (4.8)**	**0.9483 (0.0082)**	**0.8712 (0.0093)**	**0.9082 (0.0080)**	**0.95**

(c) Task 3 - Classification of true versus closely-related negative triples. Relationship type: 'Is_a'.

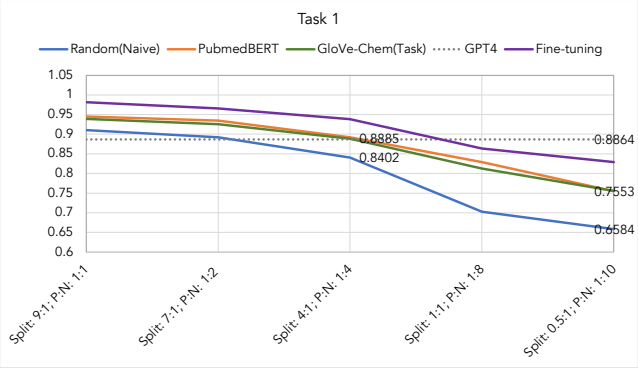
but the first setting (Split: 9:1; P:N: 1:1). For task 2, however, GPT-4 never surpassed ML-based or fine-tuned approaches.

4 DISCUSSION

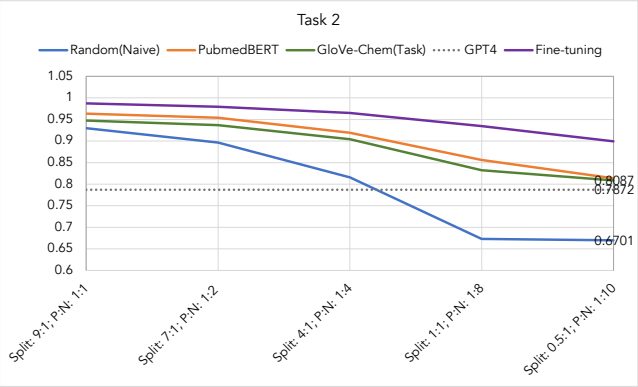
In this study, we followed three typical NLP paradigms, implemented a number of models using eight pretrained models, and conducted a series of experiments on three knowledge enrichment tasks. These have generated comprehensive sets of results and revealed some insightful findings.

In the head-on-head comparisons, the state-of-the-art foundation models didn't perform well in our three tasks. The best NLP paradigm seems to be supervised learning methods using domain/task related pretrained distributed representations. Fine-tuning pretrained language models also performed strongly, particularly in task 2. However, such interpretations might only translate to situations where there is sufficient training data for the task on hand, as the ML models and fine-tuned models were trained or fine-tuned on plenty of data (at the scale of hundreds of thousands of triples). Further experiments simulating five data availability scenarios revealed more detailed and practical insights. For tasks 1 and 3, GPT-4 was clearly superior when the training data contained no more than 6,000 triples with an imbalance around 1:8 (positive:negative) or higher. However, GPT models seemed particularly poor in task 2 (i.e., classifying wrong relationship directions), where their in-context learning capacities never surpassed other NLP paradigms in all five scenarios tested.

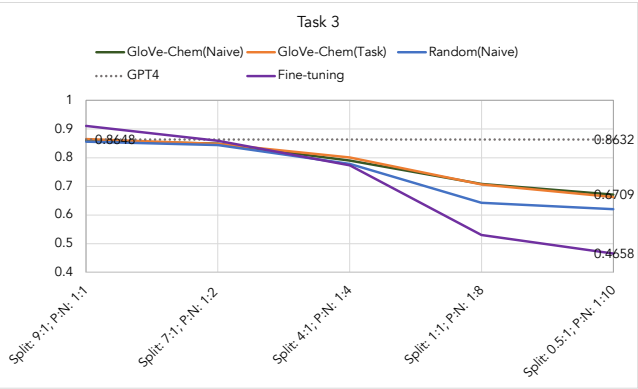
Among the three GPT models, the domain specific BioGPT was not as good as generic counterparts. Recall scores were particularly poor. It also tends to give irrelevant answers even when prompted not to do so. This may have been due to the significant differences of training corpus size and number of parameters, and also the fact that it was not further improved via reinforcement learning from human feedback. GPT-3.5 and GPT-4 also showed very consistent results reflected by their Kappa scores. Prompting these two models not to make a classification when unsure led to considerably high performances (F1 scores: 0.79-0.91) for those triples where a definitive classification was made. Combined, these observations indicate that state-of-the-art foundation models could be very promising tools for knowledge curation, albeit leaving a 5-11% of the data unclassified.



(a) Task 1. Top two most consistent ML models were those using PubmedBERT embeddings and GloVe-Chem embeddings.



(b) Task 2. Top two most consistent ML models were those using PubmedBERT embeddings and GloVe-Chem embeddings.



(c) Task 3. Top two most consistent ML models were those using GloVe-Chem and GloVe-Chem.

Figure 2: F1 scores by training data volume (split) and level of imbalance (ratio of positive:negative triples) for Tasks 1-3. Graphs depict results for representative models from all three NLP paradigms. Dashed gray lines indicate GPT-4 in-context learning performances and purple lines are those of fine-tuned PubmedBERT models. Other lines are for embedding models.

Fine-tuning pretrained language models was shown to be an effective approach for enrichment tasks 1 and 2. When there was abundant data, its performances were among the best. Fine-tuned models performed much stronger compared to supervised ML models when there was only 9% of the full dataset for training, i.e., at the scale of 55,000. They were also shown to have the greatest consistency as training data availability was decreased. However, the fine-tuning approach, at least with the PubmedBERT model used in this work, seemed to bear some shortcomings regarding task 3 in our simulation experiments. Although its performance was initially strong (using 9% of the full dataset for training), its performances deteriorated much faster with a F1-score of 0.47 in the fifth scenario (training data: 3,087 triples and 1:9 imbalance ratio). The reason behind this observation is worthy of further investigation, and may lead to some interesting findings.

Our results showed that supervised learning using distributed representations was certainly still a valid NLP paradigm for knowledge curation tasks. When abundant training data was available, even ML using random embeddings could achieve very good performances, which were superior to in-context learning using GPT models. Task-specific pretrained embedding models (trained on ChEBI related articles) were clearly very useful to such curation tasks, achieving the best performances in the various setups explored. In particular, W2V-Chem embeddings - only trained on around 7,000 PubMed articles - achieved surprisingly good performances. This demonstrates the effectiveness of a simple model (word2vec) with a small task-related corpus in downstream tasks.

A key limitation of this work was that only a single ontology/KG was utilized, potentially leading to questions on the generalizability of these findings. To mitigate this, our study introduced three different types of curation tasks, and assessed model performances in five different data availability scenarios. Combined, these generated 15 different scenarios, representing a comprehensive exploration of the effectiveness of these approaches in a diverse range of settings. Nevertheless, future work using diverse datasets would produce more conclusive findings across different application domains. The other limitation was the potential reproducibility issue caused by the use of OpenAI's GPT models via their API access. It is well known that these models are continually undergoing revision and improvement. For example, our initial GPT-3.5 experiments conducted in July 2023 yielded significantly poorer results than the latest run on the same model in November 2023. Future work should evaluate the use of open source GPT models like Meta's Llama [30].

5 CONCLUSION

This work evaluated three NLP research paradigms in the context of knowledge curation for enriching biomedical ontologies with extensive experiments and in-depth analysis. We found in-context learning using the state-of-the-art LLMs did not yield the best performance. However, they do have their utilities when proper prompting strategies are used. When the training dataset size was as big as 24,000, smaller, domain-specific BERT based model can beat large, open domain state-of-the-art LLMs. Also, supervised learning approaches are not obsolete. Specifically, they outperformed LLMs significantly when there is a large enough training data.

ACKNOWLEDGMENTS

This work was supported by UK's Medical Research Council (MR/S004149/1, MR/X030075/1); National Institute for Health Research (NIHR202639); British Council (UCL-NMU-SEU International Collaboration On Artificial Intelligence In Medicine: Tackling Challenges Of Low Generalisability And Health Inequality); Iris.AI - The AI Chemist (Research Council of Norway); UCL Global Engagement Fund 2022/2023; HW's role in this research was partially funded by the Legal & General Group (research grant to establish the independent Advanced Care Research Centre at University of Edinburgh). The funders had no role in conduct of the study, interpretation, or the decision to submit for publication. The views expressed are those of the authors and not necessarily those of Legal & General.

REFERENCES

[1] Daniel M Bean, Honghan Wu, Ehtesham Iqbal, Olubanke Dzahini, Zina M Ibrahim, Matthew Broadbent, Robert Stewart, and Richard JB Dobson. 2017. Knowledge graph prediction of unknown adverse drug reactions and validation in electronic health records. *Scientific reports* 7, 1 (2017), 1–11.

[2] Charles Bettembourg, Christian Diot, and Olivier Dameron. 2015. Optimal threshold determination for interpreting semantic similarity and particularity: application to the comparison of gene sets and metabolic pathways using GO and ChEBI. *PloS one* 10, 7 (2015), e0133579.

[3] Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. 69–72.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics* 5 (2017), 135–146.

[5] Xin Cheng, Weiping Shi, Wenlong Cai, Weiqiang Zhu, Tong Shen, Feng Shu, and Jiangzhou Wang. 2021. Communication-efficient coordinated RSS-based distributed passive localization via drone cluster. *IEEE Transactions on Vehicular Technology* 71, 1 (2021), 1072–1076.

[6] Paula De Matos, Rafael Alcántara, Adriano Dekker, Marcus Ennis, Janna Hastings, Kenneth Haug, Inmaculada Spiteri, Steve Turner, and Christoph Steinbeck. 2010. Chemical entities of biological interest: an update. *Nucleic acids research* 38, suppl_1 (2010), D249–D254.

[7] Kirill Degtyarenko, Paula De Matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan McNaught, Rafael Alcántara, Michael Darsow, Mickaël Guedj, and Michael Ashburner. 2007. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic acids research* 36, suppl_1 (2007), D344–D350.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Patrick Ernst, Amy Siu, and Gerhard Weikum. 2015. Knowlife: a versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC bioinformatics* 16 (2015), 1–13.

[10] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)* 3, 1 (2021), 1–23.

[11] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (CSUR)* 54, 4 (2021), 1–37.

[12] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[13] Dimitri Kartsaklis, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Mapping Text to Knowledge Graph Entities using Multi-Sense LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 1959–1970. https://doi.org/10.18653/v1/D18-1221

[14] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2023. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications* 82, 3 (2023), 3713–3744.

[15] Yang Li, Buyue Qian, Xianli Zhang, and Hui Liu. 2020. Graph neural network-based diagnosis prediction. *Big Data* 8, 5 (2020), 379–390.

[16] Xuan Lin, Zhe Quan, Zhi-Jie Wang, Tengfei Ma, and Xiangxiang Zeng. 2020. KGNN: Knowledge Graph Neural Network for Drug-Drug Interaction Prediction.. In *IJCAI*, Vol. 380. 2739–2745.

[17] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

[18] Yingtong Liu, Junguk Hur, Wallace KB Chan, Zhigang Wang, Jiangan Xie, Duxin Sun, Samuel Handelman, Jonathan Sexton, Hong Yu, and Yongqun He. 2021. Ontological modeling and analysis of experimentally or clinically verified drugs against coronavirus infection. *Scientific data* 8, 1 (2021), 16.

[19] Jake Luo, Min Wu, Deepika Gopukumar, and Yiqing Zhao. 2016. Big data application in biomedical research and health care: a literature review. *Biomedical informatics insights* 8 (2016), BII–S31559.

[20] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics* 23, 6 (2022), bbac409.

[21] Khalid Mahmood Malik, Madan Krishnamurthy, Mazen Alobaidi, Maqbool Hussain, Fakhare Alam, and Ghaus Malik. 2020. Automated domain-specific healthcare knowledge graph curation framework: Subarachnoid hemorrhage as phenotype. *Expert Systems with Applications* 145 (2020), 113120.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[23] Pablo Moreno, Stephan Beisken, Bhavana Harsha, Venkatesh Muthukrishnan, Ilinca Tudose, Adriano Dekker, Stefanie Dornfeldt, Franziska Taruttis, Ivo Grosse, Janna Hastings, et al. 2015. BiNChE: a web tool and library for chemical enrichment analysis based on the ChEBI ontology. *BMC bioinformatics* 16 (2015), 1–7.

[24] Jeff Z Pan, Guido Vetere, Jose Manuel Gomez-Perez, and Honghan Wu. 2017. *Exploiting linked data and knowledge graphs in large organisations*. Springer.

[25] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8, 3 (2017), 489–508.

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

[27] Janet Piñero, Juan Manuel Ramírez-Anguita, Josep Saüch-Pitarch, Francesco Ronzano, Emilio Centeno, Ferran Sanz, and Laura I Furlong. 2020. The DisGeNET knowledge platform for disease genomics: 2019 update. *Nucleic acids research* 48, D1 (2020), D845–D855.

[28] Muhammad Asif Rasheed, Muhammad Nasir Iqbal, Salina Saddick, Iqra Ali, Falak Sher Khan, Sumaira Kanwal, Dawood Ahmed, Muhammad Ibrahim, Umara Afzal, and Muhammad Awais. 2021. Identification of lead compounds against Scm (fms10) in Enterococcus faecium using computer aided drug designing. *Life* 11, 2 (2021), 77.

[29] Jaleal Sanjak, Qian Zhu, and Ewy A Mathé. 2023. Clustering rare diseases within an ontology-enriched knowledge graph. *bioRxiv* (2023).

[30] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[32] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*. 38–45.

[33] Honghan Wu, Minhong Wang, Jinge Wu, Farah Francis, Yun-Hsuan Chang, Alex Shavick, Hang Dong, Michael TC Poon, Natalie Fitzpatrick, Adam P Levine, et al. 2022. A survey on clinical natural language processing in the United Kingdom from 2007 to 2022. *NPJ digital medicine* 5, 1 (2022), 186.

[34] Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. 2019. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 7346–7353.

[35] Jianbo Yuan, Zhiwei Jin, Han Guo, Hongxia Jin, Xianchao Zhang, Tristram Smith, and Jiebo Luo. 2020. Constructing biomedical domain-specific knowledge graph with minimum supervision. *Knowledge and Information Systems* 62 (2020), 317–336.

[36] Rui Zhang, Dimitar Hristovski, Dalton Schutte, Andrej Kastrin, Marcelo Fiszman, and Halil Kilicoglu. 2021. Drug repurposing for COVID-19 via knowledge graph completion. *Journal of biomedical informatics* 115 (2021), 103696.

[37] Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. 2019. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific data* 6, 1 (2019), 52.

[38] Yixiao Zhang, Xiaosong Wang, Ziyue Xu, Qihang Yu, Alan Yuille, and Daguang Xu. 2020. When radiology report generation meets knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 12910–12917.