Taylor & Francis
Taylor & Francis Group

Expert Opinion on Drug Discovery

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/iedc20

Natural language processing in drug discovery: bridging the gap between text and therapeutics with artificial intelligence

Christine Ann Withers, Amina Mardiyyah Rufai, Aravind Venkatesan, Santosh Tirunagari, Sebastian Lobentanzer, Melissa Harrison & Barbara Zdrazil

To cite this article: Christine Ann Withers, Amina Mardiyyah Rufai, Aravind Venkatesan, Santosh Tirunagari, Sebastian Lobentanzer, Melissa Harrison & Barbara Zdrazil (30 Apr 2025): Natural language processing in drug discovery: bridging the gap between text and therapeutics with artificial intelligence, Expert Opinion on Drug Discovery, DOI: 10.1080/17460441.2025.2490835

To link to this article: https://doi.org/10.1080/17460441.2025.2490835

Published online: 30 Apr 2025.

EXPERT OPINION ON DRUG DISCOVERY
https://doi.org/10.1080/17460441.2025.2490835

Taylor & Francis
Taylor & Francis Group

REVIEW

OPEN ACCESS Check for updates

Natural language processing in drug discovery: bridging the gap between text and therapeutics with artificial intelligence

Christine Ann Withers [a*], Amina Mardiyyah Rufai [b*], Aravind Venkatesan[b], Santosh Tirunagari [b], Sebastian Lobentanzer [c,d,e], Melissa Harrison [b] and Barbara Zdrazil [a,e]

[a]Chemical Biology Services, European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, UK; [b]Literature Services, European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, UK; [c]Institute of Computational Biology, Helmholtz Centre, Munich, Germany; [d]Faculty of Medicine and Heidelberg University Hospital, Heidelberg University, Institute for Computational Biomedicine, Heidelberg, Germany; [e]Open Targets, European Bioinformatics Institute (EMBL-EBI), Cambridge, UK

ABSTRACT

Introduction: The field of Natural Language Processing (NLP) within the life sciences has exploded in its capacity to aid the extraction and analysis of data from scientific texts in recent years through the advancement of Artificial Intelligence (AI). Drug discovery pipelines have been innovated and accelerated by the uptake of AI/Machine Learning (ML) techniques.

Areas covered: The authors provide background on Named Entity Recognition (NER) in text – from tagging terms in text using ontologies to entity identification via ML models. They also explore the use of Knowledge Graphs (KGs) in biological data ingestion, manipulation, and extraction, leading into the modern age of Large Language Models (LLMs) and their ability to maneuver complex and abundant data. The authors also cover the main strengths and weaknesses of the many methods available when undertaking NLP tasks in drug discovery. Literature was derived from searches utilizing Europe PMC, ResearchRabbit and SciSpace.

Expert opinion: The mass of scientific data that is now produced each year is both a huge positive for potential innovation in drug discovery and a new hurdle for researchers to overcome. Notably, methods should be selected to fit a use case and the data available, as each method performs optimally under different conditions.

ARTICLE HISTORY
Received 28 November 2024
Accepted 4 April 2025

KEYWORDS
Drug discovery; Natural language processing; named entity recognition; large language model; knowledge graph; machine learning; deep learning; ontology

1. Introduction

Drug discovery is an inherently complex and knowledge-intensive process, involving large volumes of scientific literature, clinical trial data, patents, electronic health records, and other textual resources. Given the tremendous pace of growth of such data and the need for integrating information from multiple sources and even multiple scientific domains, it is wise to automate and standardize the data retrieval processes in order to make them reproducible. Moreover, rather than analyzing unstructured textual data, an approach that efficiently delivers structured information – including a standardized semantic understanding of such information – is desired.

Before the widespread use of Natural Language Processing (NLP) techniques, information extraction from large amounts of text was carried out using simpler, rule-based, and/or statistical methods. Examples include the use of keyword-based, Regular expressions (RegEx)-based and rule-based pattern matching systems [1]. These methods often depend on expert-defined rules to identify or infer relevant information in a text. For the statistical methods used in this area, methods for ranking and identifying the most relevant terms in large collections of text such as TF-IDF (Term Frequency-Inverse Document Frequency) are worth mentioning [2]. Likewise, clustering (such as K-means clustering) [3] and Topic-modeling approaches (such as Latent Dirichlet Allocation (LDA)) [4] provide the statistical basis for automatically grouping documents or terms that are semantically similar without requiring a deep linguistic understanding. While all these methods still have their place in Drug Discovery and related fields, they generally work better for well-defined tasks, and they would require more manual intervention, e.g. for defining rules to be applied. However, these methods are all known to lack context understanding. The handling of synonyms, for instance, becomes a cumbersome manual task. Therefore, such earlier methods have somewhat limited scope.

NLP is a field of artificial intelligence (AI) which aims to develop computational methods that make human language understandable and interpretable by computers. Within the life sciences, NLP has become vital when working on drug discovery research and target identification [5,6]. Using NLP methods can allow us to gain an understanding of, e.g. the

CONTACT Barbara Zdrazil bzdrazil@ebi.ac.uk Chemical Biology Services, European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, Cambridgeshire CB10 1SD, UK
*These authors contributed equally.

Article highlights

- Natural Language Processing (NLP) methods are crucial for drug discovery, enabling the extraction of insights from vast biomedical literature. NLP helps identify drug targets, relationships between diseases and proteins, and aids in drug repositioning, as seen with the repurposing of baricitinib for COVID-19 treatment.
- Various NLP techniques, including Named Entity Recognition (NER), relationship extraction, and text classification, are used to process unstructured biomedical data. Recent advances, particularly in deep learning (DL) and Large Language Models (LLMs), have significantly improved NLP capabilities, though challenges in reliability, interpretability, and computational requirements remain.
- Ontologies and Knowledge Graphs (KGs) are key tools in organizing and interpreting biological data. KGs store complex relationships between biological entities, and embeddings from models like BERT enable better contextual understanding of data. Combining KGs with machine learning can help uncover new patterns and relationships in drug discovery, improving efficiency and outcomes.
- Traditional methods like string-matching and rule-based approaches for relationship extraction have limitations in scalability and context awareness. Machine learning (ML) models, while improving recall and handling complex relationships, still face challenges like false positives, data quality requirements, and the need for expert-driven feature engineering.
- Deep learning models, particularly Transformer architectures like BERT, have improved the ability to capture complex contextual relationships in biomedical texts, enhancing performance in tasks like Named Entity Recognition (NER). However, these models rely heavily on large, high-quality annotated datasets and still struggle with long-range dependencies and data scalability in the context of drug discovery.
- GPT models are versatile, autoregressive language models that excel in text generation and task adaptability through instruction fine-tuning, with applications in drug discovery, but they face challenges like 'hallucinations' and data limitations, which can be mitigated by techniques like RLHF, RAG, and specialized prompting or fine-tuning.

known state of biology surrounding a disease or drug target of interest. This processing of biological corpora can help to find and strengthen confidence in relationships between diseases and potential drug-targets, those being specific molecules in the body (most commonly proteins), as well as investigate the underlying reasons why two concepts are linked. For example, shared links to a biological mechanism of both a protein and a cell type can inform researchers on the context in which a given protein plays a role in a mechanism.

However, in order to extract this information from a text corpus, the drug discovery researcher needs to implement a complex chain of technical steps. The corpus needs to be tokenized (transforming words into elements amenable to NLP), sentenced (splitting these tokens into manageable segments), normalized (e.g. with respect to upper and lower case), and lemmatized (reducing inflections of words to their stems; e.g. 'involved' and 'involving' to 'involve'). These technical representations of words then need to be processed to annotate them semantically; the researcher may need to determine the function of each word ('part-of-speech'), their dependencies ('what is the object of the verb'), and determine their semantic roles ('who does what to whom'). Finally, the researcher may want to map entities, such as cell types and molecular mechanisms, to existing concepts (potentially from ontologies), which involves named entity recognition (NER), concept normalization or named entity linking (NEL), and extraction of events or relationships. Additional tasks can involve document-level annotation and classification, clustering, summarization, disambiguation, and coreference resolution. Once all these technical steps can be performed with reasonable accuracy, the NLP pipeline can process text for the researcher automatically.

With this confidence in our understanding of potential disease – drug – target links, we can proceed to develop new drugs with sound backing of evidence indicating that they will be more likely to succeed in clinical trials. Extending this further, this type of data can aid in the potential repositioning of drugs, helping to repurpose approved drugs to fit new therapeutic use-cases. A pertinent example of drug repositioning for success would be the massive efforts that happened globally at the outbreak of the COVID-19 pandemic [7]. Bioinformatics methodologies including Machine Learning (ML)-based NLP and knowledge graph (KG) generation were used to demonstrate the promising idea of repositioning the drug baricitinib, which had been approved to treat rheumatoid arthritis, to treat hospitalized patients with COVID-19 [8]. These initial assessments led to the successful approval and use of the drug to treat COVID-19 [9] and this is but one of many examples of drug repositioning success during the pandemic.

Evidence supporting a candidate drug target or drug being functionally tied to a disease matters because the odds are stacked against clinical trials succeeding. Only around 13.8% of all trials make it through phases I-III to approval [10]. The exponential increase in the volume of biological data now being produced has created huge potential for knowledge to gain insights from. However, in consequence, manually curated ontologies can no longer match the pace of publications, mainly due to the lack of flexibility in these earlier means of Named Entity Recognition (NER). While manually curated ontologies have their place, they remain very precise in their core tasks and can be used to train ML models [11], instances of novel concepts that have not been curated into ontologies can be missed. In drug discovery, there is importance in striking a balance between acknowledging well-connected concepts and being agile enough to link novel connections as they appear in research.

In this review, we present an overview of NLP techniques currently available to drug discovery professionals, including information retrieval, Named Entity Recognition (NER), relationship extraction, and text classification. Each of these approaches addresses specific challenges in processing and interpreting unstructured biomedical text, facilitating the identification of novel drug candidates, mechanisms of action, and safety profiles. Further, we evaluate advances to these principles via deep learning (DL) methods. Deep learning is a subset of machine learning, and utilizes multi-layered neural networks to analyze data hierarchically, uncovering complex patterns that traditional machine learning methods – limited to simpler data representations – often fail to detect. Recently, transformative technologies in NLP, particularly Large Language Models (LLMs), have driven substantial improvements across various NLP tasks. Current trends highlight the impressive capabilities of LLMs in handling large-scale, unstructured biomedical data, making them promising tools for drug discovery. However, questions arise about their practical integration to sensitive and high-stakes tasks, such as

decision-making in drug discovery, where reliability and factuality are critical.

Throughout this review, when referring to LLMs, we will generally be referencing current GPT-like models unless stated otherwise. Both BERT-like and GPT-like models originate from the Transformer architecture, which is briefly discussed in this article. BERT-like models, considered early LLMs, use only the encoder component of Transformers and are widely used for applications requiring semantic and syntactic understanding of text, such as text classification or entity recognition. They are often referred to as 'representative models' because they focus on text comprehension rather than generation. On the other hand, GPT-like models are decoder-only architectures optimized for generative tasks.

We explore these differences in terms of their underlying design and the ways they process and represent information, elucidating their respective strengths and limitations in the context of drug discovery. In addition, Generative models are discussed extensively in a dedicated section of the article.

All traditional and deep NLP methods carry both benefits and limitations, with each suited for different use-cases. For example, while rule-based systems and conventional machine learning methods are often interpretable and domain-specific, they may lack the scalability and adaptability offered by deep learning approaches. Conversely, LLMs, despite their impressive capabilities, require substantial computational resources and are prone to challenges such as hallucination and interpretability issues.

Finally, we will evaluate the landscape of NLP in drug discovery, highlighting both established methods and cutting-edge advancements. By analyzing the capabilities and limitations of existing approaches, we identify opportunities where emerging technologies can address current gaps, thereby enhancing the utility and accessibility of NLP tools. Addressing these challenges has the potential to significantly streamline the preclinical development pipeline, reducing both time and costs associated with drug discovery. Moreover, by enabling more precise and efficient analysis of biomedical data, NLP can contribute to improved therapeutic outcomes, ultimately advancing the broader goal of delivering better healthcare solutions to patients.

2. Methodology for the review

Literature was searched through Europe PMC, ResearchRabbit and SciSpace. We then accessed exclusively literature that was either open source or in keeping with EBI license agreements. We adhere generally to the rule that the most up-to-date and valuable AI/ML-related literature is ~10 years old max. However, there is a requirement to reference foundational papers regardless of their publishing date, due to the precedents they set.

3. Using ontologies for named entity recognition

More traditional means of recognizing and uniquely categorizing terms in textual data include the use of vocabularies containing terms, alongside any synonyms of said term, linked together by relations to create ontologies. These definitions of terms and relations defined in ontologies can be thought of as trees, a subtype of graphical structures [12], meaning they are easily translatable into Knowledge Graphs [13]. An example of this from the Experimental Factor Ontology (EFO) demonstrates these concepts. Consider the entity 'lung,' with the unique identifier (UID) UBERON_0002048 [13,14]. This entity is a subclass of the 'respiratory system' entity, with many cell types, such as 'lung secretory cell,' being linked to it through a relation named 'part of' (Figure 1). Unique identifiers (UIDs) are commonplace in NER – using ontologies to allow cross-referencing of different ontological sources. In addition to the ability to 'ground' or normalize terms found in-text to unique concepts, UIDs are integral in many downstream processes like KG generation. Strictly defining concepts in this manner allows for strong compliance with the FAIR principles (Findable, Accessible, Interoperable, Reusable), encouraging open source and interoperable data among scientists [15]. Another benefit of ontologies is the ability to find crossovers between identified entities through the graph-like structure that an ontology provides. This can prove useful when trying to assess data relating to hypothesis-driven questions a researcher may wish to answer. An example of such a question may be: 'what association is there of a given disease to a given gene?' Downstream analysis of data collected through NER, be that via ontologies or NER models, often still requires the grounding of identified terms in text.

Using ontologies via NER can be considered as literal string-matching of terms which map to UIDs in texts and, although NER via ontologies provides a very precise means of identifying terms, unless there is an introduction of 'rules' around entity annotation in-text there is a risk of false positives (incorrectly tagged terms) being introduced. Rules in this case may include the use of no-go terms, where if an entity hit is found but this hit co-occurs with a no-go term in a range of 'x' words, that hit would be excluded. If we were interested in annotating proteins like the ataxia-telangiectasia mutated (ATM) protein for example, we could theoretically run the risk of incorrectly annotating a mention of a bank ATM. This problem of capturing false positives still happens with NER models, but the hope is that models are more context-aware. Ambiguity in definitions within ontologies can also be hard to align on, at times rules categorizing certain terms can be hard to maintainfor example, defining the edge 'is_a' between entities is very open ended and can lead to disagreement amongst curators [13].

It is often possible, especially using well-curated and long-standing ontologies, to link a string in a piece of text to an ontological term. However, when found entities cannot be grounded to an ontology, because a corresponding concept does not exist, we lose information that decreases recall – the fraction of true positives recovered over all true positives – leading to a lack of coverage of novel concepts or new trends in research [16]. Biomedical terminology is extensive; it has been recorded that diseases and syndromes alone account for over 220 million entities identified in text. Approximately 76.6% of these entities can be grounded to entities in the EFO, however, this still represents only 7.6% of all unique labels found in text. The rest represent a huge amount of

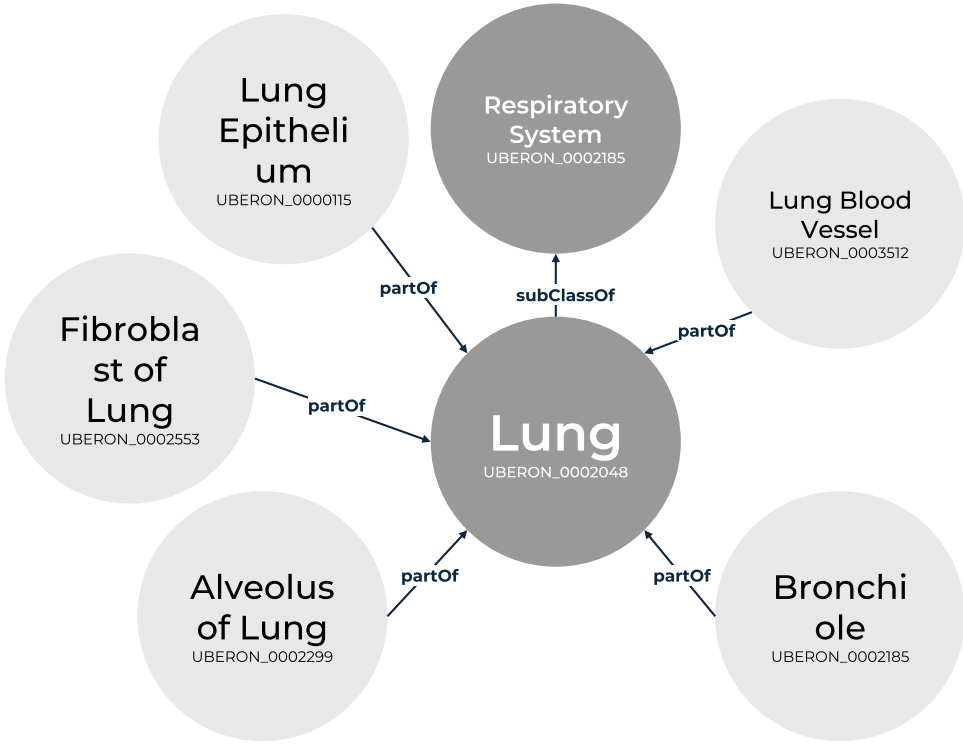

Figure 1. Illustration of the 'lung' entity sourced from EFO, alongside its parent class 'respiratory system' and a sample of cell types which are related by the 'partOf' relation to the entity 'lung.' Unique identifiers for these entities are sourced from Uberon, exemplary of importing source information from one ontology into another. The EFO also demonstrates cross-referencing where an entity such as 'lung' has references to multiple ontologies including the medical subject headings (MeSH) and NCI thesaurus ontologies.

heterogeneous, infrequent labels that escape normalizing through string-matching to ontologies [17]. This risk of missing terms increases with the volume of scientific texts that are now produced. It has been estimated that the annual growth rate of published literature across the four largest publishing databases is 4.1% [18]; with this new methods need to strike a balance between confidence in the identification of terms and the discovery of new scientific findings [19].

4. Relationship extraction from text

Beyond the scope of NER in drug discovery research, scientists often want to associate entities to one another following their identification in-text in order to discern relationships between them. Relationship extraction from text can be as simple as collecting data on the co-occurrence of named entities of a given type. Using co-occurrence works on the assumption that if particular entities occur more commonly in a sentence together than others, they are more likely to be related. The teams at Europe PMC and Open Targets – a public–private partnership for providing gene-disease associations together with their sources of evidence – are using drug target – disease co-occurrence in the literature as evidence for association between entities [17]. For example, if a protein is found to be more likely to co-occur in-text with a disease, then it could be assumed that the protein plays a part in some aspect of the disease. A simple demonstration of the use of co-occurrence of entities to derive relationships is GoGene, a web-interface wherein users enter a search term and receive a list of genes, identified through the Gene Ontology, prioritized by co-occurrence of the gene with the grounded search term entities [20,21].

Whilst co-occurrences are measurable and simple to use, relying on co-occurrence alone depends on strong assumptions. Entities mentioned together in an article could be entirely unrelated or the text could even negate any relationship existing between them. This weakness can be partly remedied by part-of-speech (POS) tagging. POS tagging involves using parsers to identify POS: nouns, verbs, adjectives and adverbs. Once identified, these can be used to find triples of words, most commonly, 'Subject-Verb-Object' triples. POS-tagging and focus on terms of interest can build patterns with entities such as protein – protein interactions, including searching for a pattern such as [protein]–[bind*]–[protein]; bind* being mapped to all words of the stem, i.e. binds, binding, etc. These efforts allow for collation of protein binding triplets and the subsequent reconstruction of protein – interaction pathways [22]. These triples can also be annotated with negation. For example, the edge [protein]–[bind*]–[protein] can become [protein]–[not*bind*]–[protein]. Extracted relationships between entities styled in the triple format suit ingestion into KGs well.

5. Knowledge graphs in natural language processing

In the age of rapid expansion of scientific data, it is paramount that researchers correctly store and handle data. KGs are a way to store information as graphs (nodes and edges) and are often both machine- and human-readable. Technically, they

can take the form of labeled property graphs, as implemented by Neo4j [23]. Neo4j uses a specialized graph query language, Cypher, optimized to retrieve information from graph-structured databases. Benefits of KGs in comparison to tabular-based databases such as SQL include the lack of need to use foreign keys and JOIN operations [24] between tables to extract relationships from data. For highly connected data, KGs are far more usable and can extract more complicated relationships between data compared to tabular structures. A KG's structure is defined by schema, as shown in (Figure 2), defining the relationships between the entity types present in the graph. Similar to ontology structures, a KG schema will contain 'nodes' (or 'vertices') connected by 'edges' (or 'relations'). In the case of the KG shown in Figure 2, we have a simplified definition of entity types and their relationships pertaining to the central dogma of molecular biology [26]. Entities of a given type, such as Glucose-6-phosphatase which is a 'protein,' would be related to relevant entities via edges. An example here would be the triple (Glucose-6-phosphatase)–[HAS_FUNCTION]–(gluconeogenesis), with gluconeogenesis being a 'molecular function' entity.

While Figure 2 is a simple demonstration of nodes being linked by edges, a large number of details can be added to this pattern. We can consider 'protein' entities and their link to 'molecular function" entities – these molecular functions will be part of biological processes. Proteins will be part of biological pathways and these pathways linked to phenotypes of disease. All of this can be added to this KG's schema together with detailed metadata for each entity type, such as data provenance, cross-references from external databases, and experimental results. KGs hold the potential to embody aspects of the massive complexity of biology, facilitating the discovery of trends in the aggregated data. KG schemas are important to consider from a design perspective as their efficient design enables the most high-quality insight to be gained from the data available. Methods to then normalize data to a defined schema can be chosen appropriately. More traditional dictionary-based approaches of normalization may be considered sufficient where data is consistent; for the most part dictionaries are still regarded as an apt means by which to standardize and consolidate data from multiple sources [27]. Novel methods to assist in this generation of data for KGs include the use of models, which can define entities in text which are not found in predefined dictionaries as well as the notion of hierarchical classification models which allow for more fine-grained NER. More granular attempts at term classification aim to remove ambiguity within captured terms [28].

Relationships between two entities, such as the aforementioned co-occurrences in text or identification of subject – verb – object triples, can be used to add weights to KG edges. Additional evidence can be used to determine the strength of a link between two entities; the ability to use multi-faceted data – including scientific texts, clinical trial outcomes, experimental results, and genomic sequencing data – casts a wide net over what can be learned about subject areas of interest through KGs. However, this abundance of data remains to be an issue. Molecular genomics data alone is expected to exceed four exabytes in 2025 [29] and patient level data are also becoming more and more available to researchers through medical research programmes and wearable technologies [30].

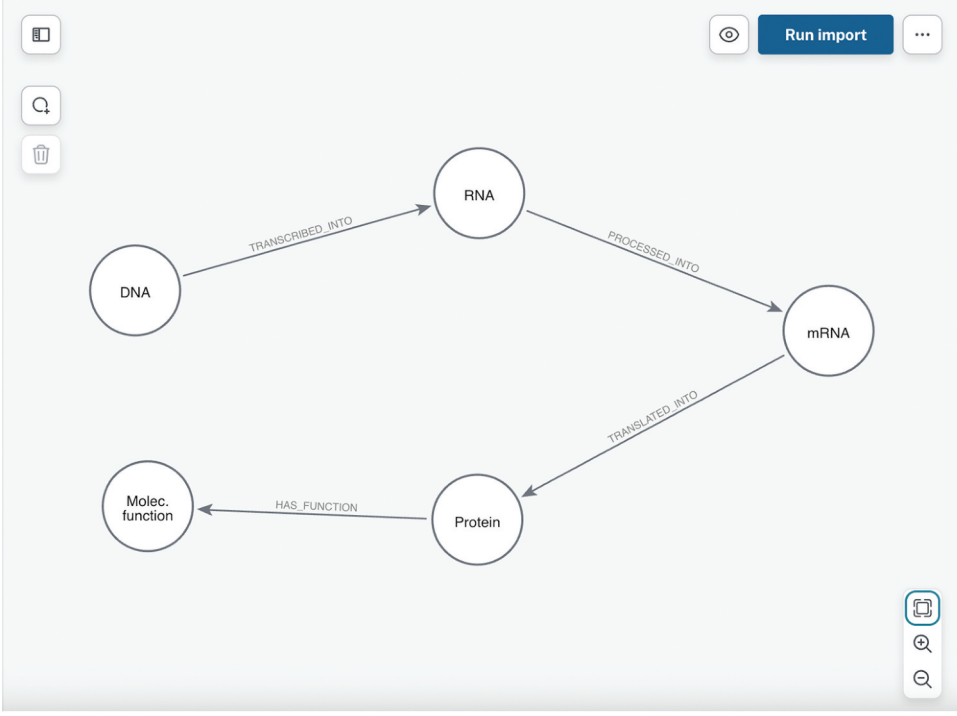

Figure 2. Demonstration of a knowledge graph schema describing the central dogma of molecular biology (simplified). This schema was instantiated in Neo4j's aura console user interface and sets the scene of how a KG is to be planned out, with data then being imported to comply with the defined schema. Neo4j aura DB console accessible at [25]. Screenshot used with permission of Neo4j.

Utilisation of KGs for uncovering patterns in big data can include network inference of relationships through ML models [31], for instance via BioBLP (BioBERT for Link Prediction), a flexible framework overlaying multimodal data so that different sources can complement each other and lead to novel insights [32]. With BioBLP, data are handled appropriately in order to be represented as embeddings. Embeddings are vectors or numbers which represent a piece of data in a lower-dimensional space. Each number in a vector represents some aspect of the data point it is representing and with embeddings being quite large vectors, they can hold features such as contexts in which a data point is seen. We assume that more similar concepts would appear more closely together in vector space.

Embeddings can be categorized based on the type of data they encode. In NLP, embeddings typically take the form of word or sentence embeddings. Word embeddings represent individual words or tokens, focusing on word-level relationships. Older methods used to generate these types of embeddings include Bag-of-words (BoW), Word2Vec [33] and GloVe (Global Vectors for Word Representation) [34]. These methods typically used static embeddings, meaning that each word had a fixed representation regardless of the context in which it appears. One key limitation here was inability to account for polysemy, the phenomenon where words have multiple meanings depending on the context. For example, the term 'receptor' in drug discovery can refer to a biological receptor (e.g. a protein or molecule involved in signal transduction) or a drug receptor (the target that binds to a drug molecule to produce its effect). In this example, Word2Vec would produce the same vector for the word 'receptor' regardless of whether it means different things in both contexts. Conversely, newer models like BERT (Bidirectional Encoder Representations from Transformers) and other transformer-based architectures (discussed later in the text) produce contextualized embeddings, where the representation of a word changes depending on its surrounding words. Word embeddings are commonly used in tasks such as word similarity, analogy, and part-of-speech tagging.

On the other hand, sentence embeddings capture the meaning of entire sentences, encoding the broader relationships between words in a sentence. These embeddings account for both syntactic structure and the semantic interactions of words to understand how they function together in context. Unlike word embeddings, which focus on individual words, sentence embeddings reflect how the sequence of words and their relationships contribute to the overall meaning of the sentence. This makes them especially useful for tasks requiring a holistic understanding of text, such as sentence classification, question answering, and relationship extraction [33].

Traditional methods for generating sentence embeddings shared similar limitations to those seen with word embeddings, particularly in fully capturing the complex and contextual nature of language. Common approaches included BoW, TF-IDF (Term Frequency-Inverse Document Frequency), and averaging word embeddings. These methods typically generated sentence-level embeddings by aggregating word-level embeddings, often leading to oversimplification of the semantic and syntactic relationships in the sentence. While techniques such as TF-IDF are still widely used due to their simplicity and interpretability, they fall short in capturing deeper contextual meanings and relationships within a sentence.

Since NLP primarily focuses on textual data, recent approaches often employ newer embedding models such as BERT [35] or BioBERT [36] for the advantages they offer in processing sequential data. Conversely, other data formats, such as images, are better handled by deep learning architectures like Convolutional Neural Networks (CNNs). CNNs are designed to process spatial data and are particularly effective at recognizing patterns and features in visual data, such as protein structures, 2D or 3D molecular structures, and microscopy images used to analyze cellular responses to drug treatments. While CNNs are well-suited for these tasks, some of them, like protein structures, can also be processed by transformer models when adapted to handle such tasks. A good example here is AlphaFold [37].

Once embeddings are generated, they are aggregated to form a unified representation of an entity. This can be done through simple concatenation or by averaging the embeddings. A graph consisting of embedded entities can then be interpreted using statistical functions like cosine similarity or Euclidean distance [38], which help assess the relative similarity between entities. For instance, comparing the embeddings of the diseases 'heart disease' and 'coronary thrombosis' would likely show them as more similar to one another than to a condition like 'epilepsy,' as the embedding structure captures subtle relationships that go beyond mere word matching.

Koscielny et al. integrated data from the Open Targets platform [39] with clinical trials outcomes in a KG representing drugs, genes and diseases. Then, using a tensor factorization model, the group represented links between genes, diseases and clinical trial results – this included the prediction of missing pieces of tensors. A major outcome of this paper showed that enriching the tensor factorization model with KG embeddings led to improved predictions of drug target pairs [40]. KGs and LLMs can interface both ways – as shown by Open Targets, KGs can be used to induce factuality in LLMs and LLMs can be used to produce KGs, highlighting their complementary roles in advancing biomedical knowledge discovery. A dedicated section later in this article delves deeper into the interplay between LLMs and KGs, exploring their interoperability and applications in drug discovery.

6. Machine learning models for natural language processing

The limitations of using older methods such as string-matching to ontologies and rule-based approaches for relationship extraction have often become too cumbersome to consider using these methods alone following the introduction of machine learning into NLP. For the moment, one irreplaceable aspect of utilizing ontologies for NER is the assignment of unique identifiers to tagged terms. Grounding terms to ontologies in this way makes the data more FAIR for collaboration across working groups as well as more interoperable for downstream data analysis and integration [15]. For now, high confidence in NER-tagged terms can be ensured

when they are grounded to a curated ontology. Oftentimes model-derived tagging is deemed as accurate, with extra trust in the tagging being confirmed by the grounding of a term to an ontology.

False positives can arise when using ontologies, given that string matching in this way lacks context awareness. By using ML models, we hope to contextualize terms to their textual surroundings, reducing ambiguities that would lead to false mappings in the traditional approach. In comparison to predefined terms, it is observed that ML models can have a lower precision, a reduced number of true positives categorized by the model overall, and much higher recall [41]. A large factor in ensuring that a model performs to the best of its ability is to ensure high-quality annotated data is used for training and validation. Annotations refer to the labels or metadata assigned to raw data, providing the model with the necessary context to learn patterns and make predictions. An example in NER would be annotating words or phrases in a sentence with their corresponding entity types, such as 'drug' or 'disease.' Complementing this process is feature engineering, where domain experts manually create and select meaningful features from the raw data to improve the model's ability to detect complex patterns. This can include transforming raw data into numerical representations, combining existing features, or applying specialized techniques to extract relevant insights. This could be in the form of creating numerical representations, combining features present in the data, or applying specialized transformations to extract insights that drive better performance in ML models. For NER, these features would be domain-specific terms, syntactic structures, and entity relationships that could help the model more accurately recognize and classify entities based on learned patterns between these relevant features and output pairs. Difficulties can always arise; humans debate on concepts being of a certain entity type. Examples here would include distinguishing between adverse events and phenotypic features of a disease.

Supervised learning models – models trained on labeled data – are commonly used for NER tasks. Biological texts are used to train the model and in these cases are manually tagged with entity types of interest. A variety of models can be adopted for this aim. Older models include conditional random fields (CRFs), first introduced in 2001 [42,43], and their predecessor, Hidden Markov Models (HMMs) [44,45]. HMMs, while effective at modeling sequential data, were soon surpassed by CRFs, as CRFs account for contextual dependencies between neighboring words more effectively, making them better suited for text mining.

As ML methods became popular in drug discovery, models such as Naive Bayes [46], Logistic Regression [47], and Support Vector Machines (SVMs) [48] became prominent for NER tasks. Both Naive Bayes and Logistic Regression are probabilistic models, but they differ in their assumptions and applications. Naive Bayes models are based on Bayes' theorem and are known for their simplicity and efficiency in text classification. They assume that features in the data are conditionally independent, an assumption that, while simplifying computation, may not always hold in complex datasets. Logistic Regression, in contrast, assumes a linear relationship between input features and the target, making it primarily applicable to linearly separable data. It remains a foundational algorithm for classification problems.

An illustrative application of ML in drug discovery is a study on cancer type classification using biomedical literature [49]. In this study, SVM, Naive Bayes, and Logistic Regression models were used to classify abstracts and full-text articles. The workflow involved preprocessing the text, extracting features using TF-IDF, and evaluating model predictions against Medical Subject Headings (MeSH) terms to assess accuracy. A key contribution of this work was the development of the SparkText framework, which leverages a Big Data infrastructure to efficiently process large-scale text data. SVM proved to be the most effective model, achieving an accuracy of 93.81%.

Table 1 summarizes the key features and limitations of the ML models discussed in this section. While all these ML models still find suitable application in less complex tasks, they are not the most capable methods available today. The potential of current state-of-the-art models is discussed in the following sections.

7. Deep learning methods for natural language processing

Feature engineering is effective in ML models, but becomes inherently restrictive when applied to the complex, large-scale biological datasets used in modern drug discovery. Feature

Table 1. An overview of key features of machine learning models showing their features and limitations.

Architectures	Type of Architecture	Key Features	Limitations
Hidden Markov Models (HMM)	Probabilistic (Graphical model)	Model sequential data, for example part-of-speech tagging	Assume observations are independent of the current state which often isn't true in language. HMMs also struggle with long-range dependencies which could include things like negation of statements in text
Conditional Random Fields (CRF)	Probabilistic (Graphical model)	Model sequential data without independence assumptions as in HMMs	Computationally expensive as CRFs look at long stretches of texts. CRFs also require large amounts of labelled data for training
Naive Bayes	Probabilistic (Bayesian classifier)	Simple and efficient; models assume feature independence which simplifies computation	Assuming feature independence can prove unrealistic in complex datasets
Logistic Regression	Probabilistic (Linear classifier)	Assumes a linear relationship between features; suited for binary classification; outputs probabilities	Limited to linear decisions; may require feature engineering with complex datasets
Support Vector Machines (SVM)	Kernel-based classifier	Capable of handling high-dimensional data spaces; capable of classification and regression tasks	Computationally expensive with large datasets; class weights must be considered if data is imbalanced

engineering is labor-intensive, requires significant domain expertise, and lacks scalability, making it difficult to capture the full complexity of biological systems. Moreover, ML models often struggle with generalization, that is, when faced with new or underrepresented data, such as novel compounds or biological targets not adequately represented in the training set, their predictive accuracy declines. This is one key limitation where deep learning (DL) models offer potential solutions. Key architectural differences between ML and DL models are illustrated in Figure 3.

DL models excel at automatically learning features directly from data. This capability arises from their use of deep neural networks, which can process high-dimensional inputs and uncover complex patterns and intricate structures within the data they are processing [51]. In the context of NLP for drug discovery, this ability enables DL models to analyze vast biomedical literature and clinical datasets, identifying associations between compounds, diseases, and molecular targets, thus aiding in tasks such as drug repurposing and candidate prioritization. Another strength is their ability to integrate multimodal data with data extracted from biological texts. Structured data linking, small molecule structures and known drug-targets to extracted concepts from text can generate more holistic insights from NLP-derived findings. These examples of structured data can inform a model on what is known to be safe and tractable in the context of target prioritization. Extrapolation from a candidate drug target to its related biological pathways can direct target identification too when considering both target safety and selectivity. Such data integration is beneficial when identifying therapeutic candidates as targets often play roles in multiple biological pathways [52,53]. Understanding the nuances of the role of various drug targets better informs researchers on the strategy to adopt in target prioritization and drug-design. Many examples of Computer-Aided Drug Design (CADD) demonstrate this [54].

In earlier DL approaches for NER in drug discovery, conventional architectures such as CNNs and Long Short-Term Memory networks (LSTMs) were widely adopted. Among these, Bi-directional LSTMs (Bi-LSTMs) combined with CRFs became popular due to their ability to model sequential data and capture relationships between words and their corresponding labels. Bi-LSTMs are a variant of LSTMs able to process input sequences in forward and backward directions. Bi-LSTMs have been explored with CRFs for NER tasks [33].

LSTMs represent one of the earliest neural language models designed to overcome the limitations of statistical models like HMMs and the vanishing gradient issues in traditional Recurrent Neural Networks (RNNs) [55]. RNNs have a memory of the sequences which they ingest. This is helpful to maintain the context of the text, however, longer sequences fall victim to vanishing gradient issues due to the fact that their gradients or 'signals' become weaker as they are repeatedly fed to the model with additional data each time the model iterates. LSTMs addressed this issue by introducing conditional memory, making them more effective at word-level processing and recognizing dependencies across sequences.

Several limitations persist with CNNs and LSTMs. CNNs, though effective at capturing local features and spatial hierarchies, struggle to model long-range dependencies and complex contextual relationships [56]. LSTMs, though benefiting from having a conditional memory that allows them to capture relationships from words situated in a sequence whether near or far, still have difficulties maintaining information across longer sequences [57]. For example, in the context of trying to understand the broader context of a drug's function across a research paper or clinical trial. These limitations have led to the growing adoption of Transformer architectures, which excel at modeling context and dependencies across entire sequences more effectively.

The Transformer architecture was introduced in 2017 by Vaswani et al. in the landmark paper 'Attention is All You Need' [57]. Transformers leverage self-attention mechanisms, enabling the model to determine the relative importance of words in a sequence independently of their positions. This

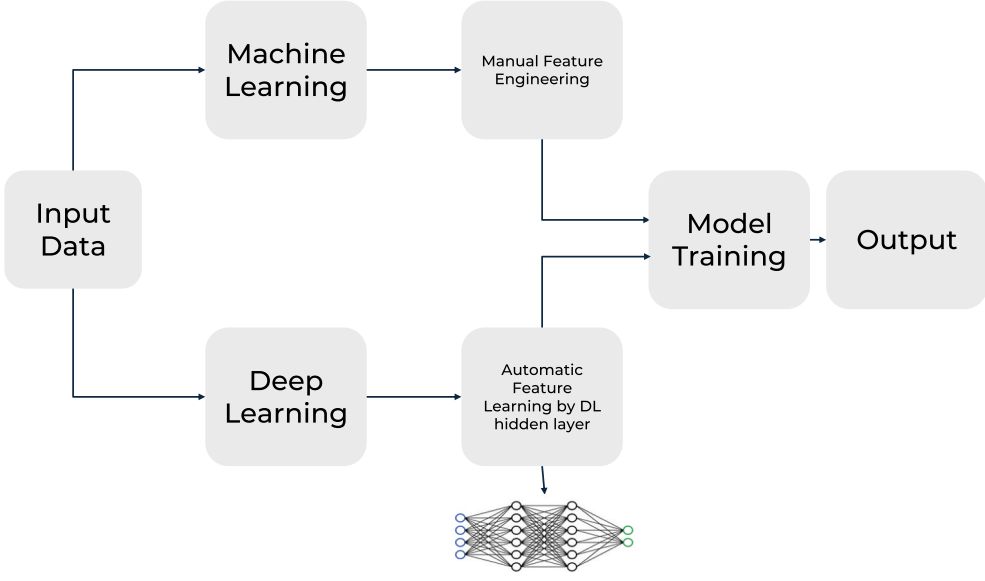

Figure 3. An illustration of the architectural difference between ML and DL. Network image reproduced from [50].

allows transformers to capture relationships across entire sequences more effectively and handle long-term dependencies more efficiently than earlier DL models [56]. One significant advantage of the transformer architecture is its ability to support parallel computation. Unlike the sequential nature of RNNs and LSTMs, which process data step-by-step, transformers can process entire sequences simultaneously, thanks to their multi-head attention mechanism. The multi-head attention mechanism, wherein multiple attention 'heads' operate concurrently, enables the model to focus on different parts of a sequence simultaneously. Each head independently learns relationships within the sequence, resulting in richer representations and faster processing. This efficient parallelization of data leads to significantly faster training and inference times compared to their predecessor, making transformers easily scalable to large datasets.

The original Transformer design is an encoder – decoder model, initially developed for sequence – to – sequence tasks, including applications like DNA or protein sequence analysis in drug discovery. Since its introduction, numerous adaptations of the Transformer architecture have been developed, including encoder-only models such as BERT [35] and decoder-only models such as GPT (Generative Pretrained Transformers) [58]. Table 2 provides a structured overview highlighting BERT and GPT models' similarities, differences and their complementary application scenarios, particularly in NLP-driven tasks for drug discovery.

The emergence of Transformers, specifically GPT, also marked a major shift in the way models are being trained and used in NLP. Traditional training strategies, which relied heavily on task-specific training and large annotated datasets are gradually being replaced by the 'pretraining/finetuning' paradigm (illustrated on the basis of BioBERT in Figure 4). In this approach, models are first trained on a massive corpora of unannotated text in a self-supervised or semi-supervised manner, allowing the model to autonomously learn general linguistic structures, including syntax and semantics [58]. The pretrained model then serves as a foundational or base model, adaptable through fine-tuning using smaller, task-specific datasets to specific target tasks such as NER. Models trained in this way are often referred to as 'pretrained language models.' One key benefit of these pretrained models is their versatility – they are not limited to a single task and in fact can be adapted to a wide range of tasks. This contrasts with traditional ML or DL methods, where models are typically trained from scratch on labeled datasets and tailored to specific tasks, such as text classification or entity recognition. These models cannot easily be reused for different tasks without undergoing the entire training process again.

In drug discovery, BERT and its biomedical variants (e.g. BioBERT, PubMedBERT) [36,59–61] have proven highly effective for NER tasks. Notably, fine-tuning on a smaller quantity of high-quality data yields significant performance gains, and the fine-tuning process typically requires less training time compared to previous DL models [35,62]. Additionally, the architecture of BERT-like models makes them particularly adept at tasks like NER due to their bidirectional attention mechanism. This mechanism is similar to how bi-LSTMs function but differs in that BERT utilizes the attention mechanism to explicitly model contextual relationships.

A piece of work utilizing NER models for the assessment of clinical trials data came from the Open Targets group in 2024 [63]. This work combined the power of two ML model styles to categorize the reasons why clinical trials were halted or failed. Firstly, a LSTM was generated to create vectors representing categories of stop reasons for clinical trialsfor example, safety issues or a lack of funding. The LSTM generation to create this network relied on the manual curation of a large set of training data, where reasons for stopping a clinical trial were first manually labeled for 3124 trials. Further manual annotation was then also required to produce a test set. The paper then proceeded with these representations of reasons for trial

Table 2. An overview of key features of BERT and GPT models showing their differences and primary use-cases.

Feature	BERT	GPT
Architecture type	Encoder-only transformer;	Decoder-only transformer;
Input processing type	Processes input bidirectionally	Uses autoregressive, unidirectional processing in a left-to-right manner
Primary application	Excel at task requiring representing texts/language	Excel at text generation tasks
Example use-cases	Biomedical NER, text classification, etc	Summarisation tasks, conversation agents, etc

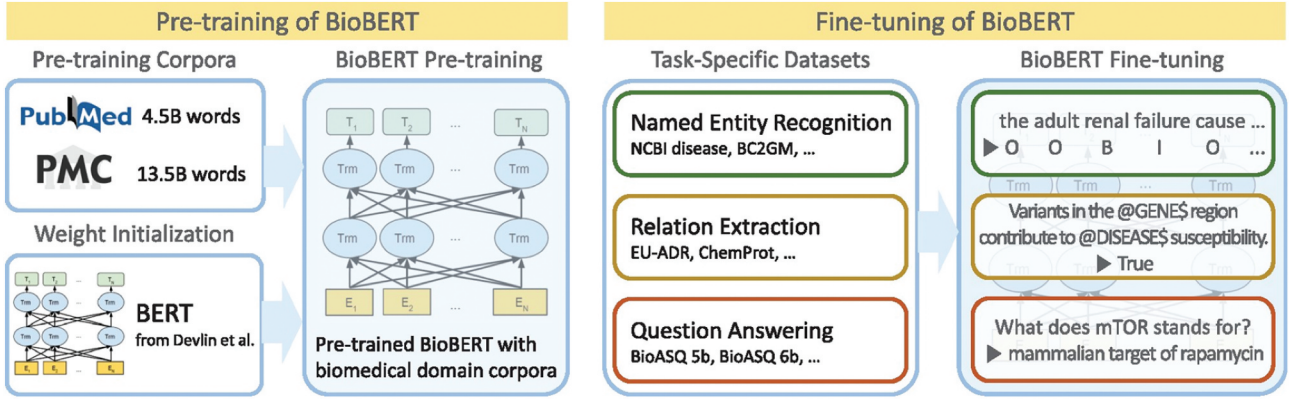

Figure 4. A depiction of the pre-training and fine-tuning workflow using BioBERT as an illustrative example. Reproduced from [36].

failure and a BERT model was specialized to categorize the listed reasons why given clinical trials failed and assessment could then be carried forward. One major finding was the halting of a given clinical trial owing to inference of a relationship between a drug target (gene/protein) lacking strong genetic associations to a candidate drug's intended disease. This backs up the consistent findings that genetic evidence supports candidate drug success in trial [64].

One persisting challenge with DL models – including transformer architectures – is their heavy reliance on large amounts of annotated data; generally, more data leads to better performance on downstream tasks. While transformer models offer many potential benefits due to being pre-trained on massive corpora, their performance during fine-tuning still depends heavily on the availability of substantial high-quality annotated datasets. Noisy or inconsistently annotated datasets can drastically reduce a model's performance.

Notably, a significant advantage of transformer architectures over other DL methods is their ability to leverage techniques such as zero-shot and few-shot learning when data availability is limited. Zero-shot learning allows models to generalize to unseen tasks without prior task-specific training by defining the output labels directly, making it suitable for prototyping tasks such as drug classification or biomedical text tagging using unlabeled data. This enables human experts to quickly assess model performance and correct errors to accelerate data annotation. On the other hand, few-shot learning guides the model using only a small number of annotated examples, helping the model understand correct input-output pairs for specialized tasks. These concepts, introduced with the release of the GPT [58], have gained widespread traction and will be explored further in the following section dedicated to 'Generative large language models.'

Nevertheless, while aiming for larger datasets is ideal, smaller amounts of high-quality data still yield better results than larger, lower-quality datasets. As Kühnel et al. [65] demonstrates on a COVID-19 NER task, BERT's performance is highly sensitive to the quality of the training data.

These data-related challenges are especially pronounced in the domain of drug discovery, where NER remains a difficult task despite advances in DL and NLP [66]. Biomedical texts are inherently complex, characterized by high length, dense relationships, and domain-specific terms. A single sentence may reference multiple interactions between different entities, and the same entity (e.g. a protein or gene) may be classified differently based on its context. Gene names can also be different depending on whether they are in a human or animal model. This makes entity disambiguation a significant challenge [67]. While pretrained language models have shown promising results, they still face notable limitations. Even when trained on a large amount of biomedical corpora, these models can still struggle to capture the nuances of complex biomedical relationships due to their fixed context windows [67]. A fixed context window defines the maximum number of tokens or words a model can process in a single input sequence. A token can either be a word, a subword, a character and can include punctuation marks; in English for example, a token is about three-fourth of a word, with 100 tokens approximately equal to 75 words. BERT-like models typically use a maximum length of 512 tokens, thus requiring long texts to be split to fit within this limit. This can hinder the model's ability to grasp cross-sentence or cross-paragraph relationships essential to understanding long – range dependencies, which are prevalent in biomedical literature. For instance, critical relationships between a drug candidate and its mechanism of action might span several sentences or an entire paragraph, necessitating the consideration of the full paragraph context to capture relevant associations.

This limitation also spurred interest in LongFormer-type [68] models, a modified transformer architecture designed to handle extended contexts more effectively. Notwithstanding, research in this area is still limited, primarily due to the increased computational demands associated with these models. While a higher context window can help capture long-range dependencies, there is a trade off with computational costs. Scalability remains a significant issue, as incorporating new entity labels into a pretrained model often requires extensive re-fine tuning on a corpus containing mentions of the new entities. This highlights another challenge where balancing data needs, computational efficiency, and the accuracy of specialized knowledge extraction is a trade-off.

Table 3 presents a summary of all the architectures discussed in this section as well as their key features and limitations.

8. Generative large language models

Recently, interest in GPT-like models has surged, largely due to the release of OpenAI's ChatGPT (GPT 3.5] [69]) in 2022. Unlike BERT, GPT models – part of the broader family of modern day 'Large Language Models' (LLMs) – are autoregressive language models, originally designed for next-word

Table 3. A summary of architectures discussed in the deep learning for drug discovery section, their features and limitations.

Architectures	Type of Architecture	Key Features	Limitations
CNNs	Classic-DL	Capture local features and spatial hierarchies; efficient for processing short input windows	Struggle with modeling long-range dependencies and contextual relationships
LSTMs	Classic-DL	Conditional memory to capture relationships in sequential data; effective for maintaining text context over time	Prone to difficulties with long-term dependencies; sequential processing limits parallelization
Bi-LSTMs-CRF	Classic-DL	Bidirectional context capture; CRFs enhance sequence labeling by modeling inter-label dependencies	Computationally intensive; limited scalability for large datasets
BERT-like models	Transformer-based	Bidirectional attention mechanism; pretrained on large corpora; adaptable via fine-tuning for tasks like NER	Context window limited to 512 tokens; challenges with long-range dependencies
Longformers	Transformer-based	Efficient attention mechanism to handle long sequences; suited for tasks requiring extended context	High computational demands; scalability challenges

prediction tasks (Figure 5). This means that given a text or sequence, a GPT model would generate the next probable word in a left-to-right manner. While this next-word prediction function mirrors older statistical language models, GPT models are different in their foundation on transformer architectures and extensive pretraining on vast corpora. Their autoregressive nature makes them particularly well-suited for text generation and completion tasks, earning them widespread recognition for impressive performance across various natural language understanding (NLU) and natural language generation (NLG) tasks.

Since their first introduction in 2018, GPT models have seen several iterations [58,69,71,72], the latest being GPT4 [73]. The release and widespread adoption of ChatGPT in both research and industry also spurred the development of similar generative pretrained language models available as either proprietary or open-source models. Proprietary models are typically accessible only through subscription-based services or paywalls, while open-source models are freely available for use and modification, sometimes subject to terms of use agreements. Notable proprietary examples include [74–76], while open-source counterparts include the LLaMA family of models [77–79], Mixtral [80,81], and others [82–84]. Many open-source models are hosted on platforms like the Hugging Face model repository [85], enabling researchers and developers to explore and adapt them for diverse applications. This trend also extends to the biomedical domain, wherein specialized variants such as BioGPT [86], BioMedGPT [87], and others [88] have been developed, reflecting a pattern similar to that seen with BERT-based models. As we will discuss later, smaller specialized models often outperform their general-purpose counterparts, particularly in domain-specific applications and tasks that require highly contextualized knowledge [48,49].

One key strength of these LLMs is their advanced NLU abilities, stemming from their specialized training to follow instructions. This in turn makes them easily adaptable to diverse tasks and domains, including highly specialized areas like drug discovery. For example, the original GPT-4 report [73] highlighted applications where LLMs can potentially streamline labor-intensive tasks, such as compound similarity analysis and chemical structure re-engineering – saving time and resources, allowing researchers to focus on high-level analysis. Other promising applications include automating data retrieval, efficient mining of scientific literature, and querying of databases for knowledge extraction.

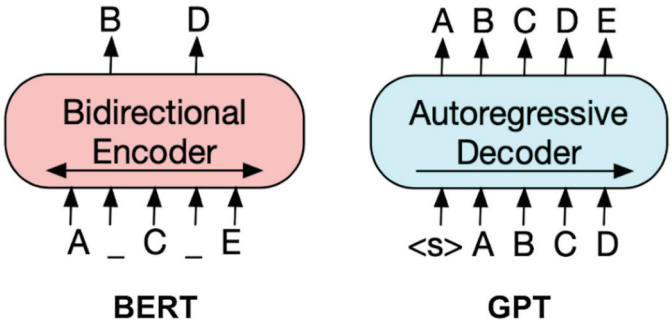

Figure 5. An illustration of the difference between BERT and GPT. BERT predicts hidden words in a sequence, while GPT uses a left-to-right transformer to predict the next possible word in a sequence. In the figure, letters A-E represent tokens, where BERT predicts masked tokens in sequences and GPT generates the next probable token. Reproduced from [70].

ChatGPT has been tested in a study involving the development of an anti-addiction drug platform aimed at cocaine addiction treatments [89]. Here, ChatGPT was integrated into a persona-based research model, with each 'persona' representing a distinct area of expertise: a drug discovery specialist, a diffusion model expert, and a coding assistant for Python-based scripting and figure generation. This work resulted in the development of a framework – 'the Stochastic Generative Network Complex (SGNC)' - which could generate multiple promising drug candidates for combating cocaine addiction. However, despite these promising results, the authors noted several inherent limitations associated with LLMs. One of which is their tendency to 'hallucinate,' meaning they may generate inaccurate or misleading information that appears factually correct. This risk is particularly concerning in fields like drug discovery, where misinformation could have severe implications. To mitigate this hallucinating effect, the researchers added an extra layer of validation to corroborate ChatGPT's outputs by cross-referencing them with expert opinions and existing scientific literature.

Broadly speaking, the validation process for LLMs can be achieved either manually – by human experts in a process known as 'red teaming' - or through advanced methods such as Reinforcement Learning from Human Feedback (RLHF) [69]. Reinforcement learning (RL) is a deep learning subfield with roots in psychology and control theory, where it was traditionally studied as an independent field. The RL approach is centered on sequential decision-making, where agents (computers or robots) learn through experience, mirroring how humans and animals learn via trial and error. What makes it different from ML or other DL approaches is that here the agent operates within an environment where it is given specific goals and encouraged to discover strategies to achieve them. For each action the agent takes toward its goal, it receives feedback: rewards for desired actions and penalties for undesired ones. In RLHF, this mechanism is adapted to LLMs by incorporating human feedback into the learning process. After the standard pretraining phase, the LLM is tasked with generating outputs for specific tasks, such as summarization, question-answering (QA), or NER. Human experts then evaluate these outputs, selecting responses that best align with desired performance. Based on this feedback, a reward model is trained to optimize and reinforce outputs that match human preferences. This reward model is subsequently used to fine-tune the LLM's responses, aligning its outputs more closely with human expectations and significantly reducing the risk of generating hallucinations or misleading information.

While RLHF can reduce hallucinations, it does not entirely eliminate them. Several factors contribute to hallucinations, one of the primary ones being a model's limited knowledge or insufficient context for a specific task [90]. Although LLMs perform well on straightforward tasks, they often struggle with complex, domain-specific tasks – such as those encountered in

drug discovery. This performance disparity arises partly because LLMs rely heavily on exposure to similar tasks or domains during their pre-training phase [90]. In scenarios where the model lacks direct or analogous pre-training data, it may attempt to fill gaps by generating outputs that appear plausible but lack a factual basis.

In addition to RLHF, other techniques are increasingly being explored to reduce hallucinations in LLMs, particularly for tasks demanding factual accuracy. Two prominent approaches are Retrieval-Augmented Generation (RAG) and KGs. Although both RAG and KGs have long been used in various computational tasks, their integration with LLMs has gained traction as a means to enhance factual grounding in critical fields like drug discovery.

LLMs can use RAG to enable real-time retrieval of up-to-date information on topics such as chemical compounds, gene targets, or molecular pathways. Typically, retrieval here refers to semantic search of an embedding space of pre-generated text embeddings, usually handled by a vector database. The quality of the retrieval (i.e. the level of relevance of retrieved information to the user's query) depends largely on the quality of the embedding algorithm applied to the input text and, to a lesser extent, on the implementation of the retrieval and reranking algorithms [91]. This retrieval capability allows the model to generate outputs grounded in current, data and task-specific knowledge, reducing the likelihood of outdated or inaccurate information and thus helping mitigate hallucinations. However, the content and nature of the provided RAG fragments has a great impact on the quality of generation. Misalignment between the model's internal knowledge and what is provided via RAG can lead to a tug-of-war between the two sources, which is an area of active research [92]. Using a multi-component system that involves decisions at various stages always harbors the danger of reasoning fragility and thus requires robust benchmarking and monitoring frameworks if applied in sensitive contexts [93].

Conversely, KGs benefit LLMs by providing structured knowledge, enhancing interpretability, and grounding outputs in known facts [94]. In drug discovery, KGs can be beneficial in that they provide contextual linkages of drugs to known adverse events, relevant biomarkers, and interactions with biological targets. For tasks like NER, this interconnected knowledge can enable LLMs to interpret complex inter-entity relationships more effectively, ultimately boosting accuracy. KGs can be used with LLMs in different ways, either by way of grounding the LLMs response to accurate information or using an LLM to construct a KG itself.

In a compelling example of integrating retrieval capabilities with LLMs [95], joint NER and relation extraction tasks were examined using two prominent LLMs – GPT-3.5 [69] and LLama-2 [78]. These LLMs were used to extract complex scientific knowledge from large datasets in materials chemistry, covering tasks such as linking dopants with host materials, cataloging metal-organic frameworks, and gathering composition, phase, morphology, and application data. This approach exemplifies the flexibility of LLMs in scientific text processing, demonstrating effective information extraction from both single-sentence and multi-sentence contexts. Importantly, the study highlighted a key advantage of LLMs over BERT-like models, which often struggle with inter-sentence relationships due to their fixed context window. Beyond simple entity extraction, the LLMs in this study generated structured outputs, such as JSON objects, that mapped intricate relationships between entities. This ability to produce structured formats like JSON or CSV on the fly is another advantage of LLMs. While this research focused on materials chemistry, the methodology has clear applications for drug discovery, where challenges like identifying drug-target interactions or mapping biochemical pathways similarly demand advanced handling of complex entity relationships within large datasets.

While this study shows the flexibility of LLMs like GPT-3.5 and LLama-2 in processing complex documents over models like BERT, it is important to note that these models do not possess unlimited context length. Instead, they leverage extended context windows to manage long-range dependencies more effectively than BERT-like models. There are also key differences in how GPT-like models and Longformers handle these extended contexts. As earlier discussed, GPT-like models are trained for next-word prediction in a causal, unidirectional manner, whereas Longformers adopt a BERT-like, masked bidirectional attention approach. While each has its own applications and limitations, a detailed discussion of both is beyond the scope of this paper.

Another notable attribute demonstrated by LLMs is their 'emergent ability' [96], which allows them to perform sophisticated tasks without explicit fine-tuning. This means LLMs can effectively handle tasks they were not specifically trained for by drawing on the extensive knowledge acquired during pre-training. This foundational knowledge is key to their success in zero-shot and few-shot learning scenarios. With the remarkable flexibility of modern LLMs, these paradigms are often implemented through prompt-based learning, where carefully crafted prompts guide the model to generate task-specific outputs efficiently. This process of carefully designing input queries or instructions to steer the model's responses toward desired outcomes is called 'Prompting' or 'Prompt Engineering.' In essence, it involves reverse-engineering what the model has learned and guiding it using specific instructions or formats to achieve the desired outputs.

Prompting or prompt engineering is the process of carefully designing and structuring prompts to guide an LLM to produce a specific, desired output for a given task. Prompting gained significant attention through OpenAI's work on GPT-2 [71] and GPT-3 [72], where it was shown to achieve impressive results in zero-shot and few-shot learning scenarios. Since then, prompt engineering has become an active area of research and a potential alternative to extensive fine-tuning. ChatGPT's release in 2022 further highlighted the versatility of prompting, showing how LLMs can handle a broad range of tasks through flexible, open-ended instructions. This flexibility not only facilitates rapid prototyping across diverse applications but also enhances accessibility, enabling even non-technical users to interact meaningfully with complex models.

In NLP, a prompt is the initial text or input given to a model to elicit a specific response or behavior. Prompts can take the

forms of natural language instructions or specialized tokens, like '[SIL]' for silence or '[UNK]' for unknown words, which direct the model to manage specific cases or focus on particular information. Natural language prompts are often open-ended and task-specific, usually crafted to work with generative models like GPT. In contrast, specialized tokens are predefined inputs designed to control certain model behaviors and are used broadly across different DL models, including transformers and LSTMs. Additionally, specialized prompting techniques, such as 'soft prompting' and 'prefix tuning,' leverage learned embeddings rather than explicit text, allowing for more nuanced and task-specific guidance of model behavior.

In the context of NER for drug discovery, LLMs can be prompted in a zero-shot manner to extract drug names or identify protein interactions from unstructured biomedical texts, without prior examples. In a few-shot setting, the process would involve providing examples that classify certain terms as drugs or illustrations of protein interactions. Once these annotated examples are provided, unlabeled samples can then be introduced, and the model is expected to generate responses that align with the established patterns.

Building on basic prompting, advanced strategies have been developed to further refine LLM reasoning capabilities. One such technique, chain-of-thought prompting [97], involves breaking down complex tasks into smaller, logical steps. For example, in drug discovery, this might involve guiding the model to first identify a biomolecule, then examine its role in a pathway, and finally predict potential drug interactions. This sequential structure could improve the model's performance on multi-step tasks by encouraging logical progression. Tree-of-thought prompting [98], introduced in 2023, extends this idea by allowing the model to explore multiple solution paths in a hierarchical manner, beneficial for tasks where biological relationships may involve multiple potential pathways or ambiguous connections.

In some cases, prompting strategies have been paired with RAG and KGs to improve model performance. In a notable example [99], zero-shot prompting was combined with a two-stage retrieval model to tackle the challenge of matching patients to clinical trials. Given the need for high factuality required for this task, the approach used a RAG module to retrieve relevant information, which was then incorporated into the prompt, guiding the model's decision-making for greater precision. Additionally, the two-stage retrieval structure improved latency, accelerating token processing and enhancing data efficiency without compromising accuracy – benefits particularly valuable in production environments where fast inference is essential.

Notwithstanding, it is worth emphasizing that designing effective prompts is a crucial aspect of fully leveraging the few-shot learning capabilities of large language models. Thoughtfully crafted and clear instructions can guide models toward generating more accurate and contextually relevant outputs. Techniques often employed include role-playing, breaking down instructions into step-by-step prompts to facilitate reasoning, problem description, or directing the model toward desired outputs by specifying formats, such as 'Return the result in 2 or 3 sentences' or 'Provide the output in JSON format.' Equally important are prompt engineering considerations for security, particularly in mitigating risks such as prompt injection attacks.

Fine-tuning LLMs on custom data is another common approach used to gain more control over the model's behavior. This method becomes particularly advantageous when there is access to quality task-specific data and a need to address privacy concerns, especially when handling sensitive information, such as proprietary data in drug development. While prompting generalist LLMs like ChatGPT can be efficient for certain downstream tasks, fine-tuning allows for models to operate in a controlled environment, making it ideal for sensitive applications. Moreover, this approach also allows for the development of specialized models tailored to specific tasks, like recognizing entities related to drug targets or cell types. In many cases, smaller, specialized models can outperform larger, generalist models.

Recently, a generalist LLM fine-tuned model derived from an earlier Google model, PaLM-2 [84] was designed to address various tasks in the drug discovery pipeline by leveraging knowledge across diverse therapeutic modalities [100]. It was trained using 709 datasets covering 66 tasks on drug discovery to equip the model to predict and process a wide variety of chemical or biological entities including small molecules, proteins, nucleic acids, cell lines and diseases. Tx-LLM shows competitive performance over generalist LLMs, achieving state-of-the-art results in 43 out of 66 drug discovery tasks, with superior outcomes in 22 of those tasks. One significant finding from this research is the evidence of positive transfer among datasets involving different drug types, indicating that fine-tuning an LLM on biological sequences has an effect on its performance on molecular datasets. The authors further argued that Tx-LLM could possibly serve as an end-to-end tool for therapeutic development, where it can allow for users to query a single model on the different steps involved in the drug development process.

Several finetuning techniques exist, with traditional supervised learning being one of the most common approaches. For LLMs, this can take the form of instruction finetuning, where the model learns from specific input – output pairings guided by instructional prompts, or classification finetuning with strictly input-to-distinct-output pairs. In instruction fine-tuning, each pairing includes instructions aiming to guide the model on how to perform specific tasks. For example, one instruction might teach the model to identify drug – target interactions, while another could focus on classifying proteins based on their functions. A sample instruction-based learning dataset illustrating this approach on a biomedical text is shown in Table 4.

In contrast, classification fine-tuning trains models for more narrowly defined tasks, where they predict categories based solely on labels seen during training. This method is effective for consistent classification tasks with structured data, such as categorizing diseases. However, classification fine-tuned models are limited to known classes, whereas instruction fine-tuned models can adapt to a wider variety of tasks, as illustrated in Table 1. Both approaches are best suited to specific use cases. Instruction fine-tuning is advantageous when the

Table 4. An example representation of an instruction-finetuning dataset on a sample snippet of text.

Instruction	Input	output	Task-type
INST: Summarise the role of Prostaglandin E2 (PGE2) in neuroblastoma progression	Neuroblastoma is the most common extracranial solid tumour in childhood and arises from neural crest cells of the developing sympathetic nervous system. Prostaglandin E2 (PGE2) has been identified as a key pro-inflammatory mediator of the tumour microenvironment (TME) that promotes neuroblastoma progression … … … … … … … … .	PGE2 acts as a pro-inflammatory mediator within the tumour microenvironment, promoting neuroblastoma progression.	Summarisation
INST: Explain the interaction between miR-574-5p and CUGBP1 and its effects on PGE2 biosynthesis in the provided paragraph		The interaction between miR-574-5p and CUGBP1 induces mPGES-1 expression, which in turn contributes to PGE2 biosynthesis in neuroblastoma cells	NLU
INST: Describe the significance of tetraspanin composition on the surface of sEV in neuroblastoma		The tetraspanin composition on the surface of sEV is associated with the function of sEV-derived miR-574-5p, suggesting that sEVs not only transport microRNAs but may also influence their functional activity in the tumour microenvironment.	NLU
INST: Extract all entities related to cell types mentioned in the text		Neuroblastoma Neural crest cells Sympathetic nervous system	Entity extraction (NER)
INST: Identify and extract key molecules and proteins involved in neuroblastoma progression from the text		- Prostaglandin E2 (PGE2) - microRNA miR-574-5p - CUG-binding protein 1 (CUGBP1) - microsomal prostaglandin E2 synthase 1 (mPGES-1)	Relationship extraction
INST: From the given text, extract relationship instances between miR-574-5p, receptors, and their effects in the TME of neuroblastoma and lung carcinoma.		- sEV-derived miR-574-5p → Acts as TLR7/8 ligand → Induces α-SMA expression in fibroblasts (Neuroblastoma TME) - sEV-derived miR-574-5p → Paracrine function → Promotes fibroblast differentiation (Neuroblastoma TME) - sEV-derived miR-574-5p → Autocrine function → Inhibits PGE2 biosynthesis (Lung carcinoma TME)	Relationship extraction

model must perform multiple tasks based on user instructions, making it ideal for chatbot applications, as seen with models like ChatGPT. On the other hand, classification finetuning is highly effective for tasks requiring precise or discriminative categorization of data.

Nevertheless, choosing between these approaches involves balancing versatility against resource constraints. Although instruction finetuning offers the flexibility to handle multiple tasks within a single model, it requires a diverse, high-quality dataset and significant computational resources. In contrast, classification fine-tuning involves less data and compute power, but its scope is limited to the specific classes the model has been trained to recognize. In both cases, data quality remains paramount, as the performance of fine-tuned models heavily depends on the quality of the annotations. This trade-off between data quality vs quantity remains persistent, as seen with previous DL models.

To address this limitation of requiring substantial data, there is growing interest in exploring synthetic data generation from small samples of real data using other LLMs. A notable example is Alpaca [101]. Synthetic data generation is a means of data augmentation which helps improve model robustness by providing additional examples that closely approximate the diversity of real data. In drug discovery, integrating human-written prompts with model-generated ones can provide significant advantages. However, careful evaluation of the generated outputs is necessary to ensure they remain evidence-based and adhere to ethical standards.

In parallel, there are ongoing efforts to create high-quality instruction-based datasets [102–104] to build more effective resources for training. These initiatives aim to provide curated, domain-specific data that can enhance model performance on complex tasks, especially in areas where traditional data collection is impractical or limited. Through these combined efforts, the field is making strides toward overcoming data-related challenges, ultimately improving fine-tuning processes for LLMs in drug discovery applications. In the context of drug discovery, the provision of domain-specific tasks may aid in the understanding of a model's logic. Opacity in LLMs can be a large blocker for their uptake in the life sciences given that at times a user cannot see *why* a model has generated a given response. Sectioning the model's work into distinct tasks can allow researchers to understand how a model makes decisions.

Table 5 provides an overview of notable open-source instruction fine-tuning datasets, including several designed for general and biomedical applications.

Table 5. An overview of some impactful open-source instruction fine-tuning datasets.

Dataset Name	Source	Domain	Key Characteristics
Alpaca	Stanford	General	Synthetic instruction-response pairs
MedInstruct	Stanford	Clinical	Synthetically generated medical instruction-response pairs
BioInstruct	University of Massachusetts	Biomedical/Clinical	Same as above
UltraMedical	Tsinghua University	Biomedical	Diverse biomedical instructions consisting of both manual and synthetic pairs

9. Expert opinion

While it is not the focus of this review article to comprehensively discuss papers that have used NLP in drug discovery, it is useful to put the described methodologies in the context of practical applications. A few such applications have been described above, such as Drug Repurposing and Target Identification, e.g. by extracting drug-target-disease relationships from text. The extraction of metadata needed to better assess the directionality of effects, or the disease relevance, such as mode-of-action information, biomarkers etc., also benefits from NLP techniques. In the area of personalized medicine, NLP techniques have proven successful for, e.g. diagnosis of genetic disease by the use of whole-genome sequencing and NLP-enabled automated phenotyping [105].

As described earlier, NLP allows setting different entities in semantic context by entity-relationship extraction (such as diseases, drug targets, proteins, biomarkers, pathways, etc.), which provides a means to weigh the importance of a specific association between related entities. Representing the extracted entities in the form of a Knowledge Graph can further serve as a comprehensive map of biomedical information helping to uncover new insights and supporting data-driven decision-making in drug discovery. Another key advantage is the possibility to extract key information from clinical trial reports such as outcomes, adverse events, dosages, and patient demographics, providing a streamlined approach to understanding trial results across multiple studies [106]. In a more prospective way, the use of NLP can help identify trends and patterns of innovation in drug discovery [107]. We can see further examples of such innovation in the demonstration of NLP models being used to screen electronic health records (EHRs) for indications of cognitive impairment in individuals; this early detection would prove incredibly positive for improving patient outcomes [108]. By mining patent data, NLP can help identify new chemical entities or new uses for existing chemicals, facilitating the identification of novel drug candidates.

In the future, it will be useful to harmonize approaches to NLP in drug discovery at multiple levels in order to help the community implement these suggestions according to best practices. Since the software and tools of the end-to-end workflow are currently fragmented, our opinion is that a dedicated tool that encapsulates this process as a software library would be most appropriate. We have previously developed knowledge management and LLM application frameworks for the biomedical domain [109]; in our most recent release, we indicate knowledge extraction as the next step for our ecosystem [93]. The particular application of the ecosystem to drug target discovery will also be driven by an ongoing project under the Open Targets consortium [110] with contributors from academia and industry. In the following, we briefly outline the tasks that will be provided under the umbrella of the planned framework.

The general idea is to encapsulate the tasks we identified above as an end-to-end application framework (working title: 'BioGather') to guide the drug discovery researcher through the sequential steps of knowledge extraction, starting from unstructured information, via the technical preprocessing, to the downstream tasks such as NER, NEL, and classification. Similar to our previous frameworks, BioGather will provide high-level access to each part of the process via a modular approach. For instance, going back to one of the introductory examples ('shared links to a biological mechanism of both a protein and a cell type can inform researchers on the context in which a given protein plays a role in a mechanism'), the framework would take the user through a guided sequence of steps to tokenize and split sentences, normalize and clean cases, remove stop words, lemmatize words into stems, parse dependencies, and apply both traditional and next-generation classification methods for NER and NEL as described in Tables 1 and 3. Each individual step can be implemented using various techniques and technical backends, which BioGather will make accessible via a unified interface.

Further, the semantic context of the knowledge to be extracted will be supplied via the established mechanism that is already used by the knowledge representation and knowledge application frameworks [91,107]. Using shared semantic definitions for the different stages of knowledge management comes with the advantage of coherent data handling and allows the user to resolve ambiguities automatically. Synchronising knowledge extraction, representation, and application via these shared definitions facilitates bidirectional synergies between the components of the knowledge management system.

The long-standing manual approaches to Natural Language Processing (NLP), which lean on human-curated ontologies for entity tagging and agreement on interpretation of results through human review, are no longer fully capable of coping with the volume of data they now must face. These methods are very human-friendly and transparent in their workings and for this reason they are not completely replaceable at present. Working groups still do need the ability to democratize data in the sense of it being FAIR. Even from the point of view of generating KGs , the need for grounding entities and cross-referencing identifiers over different data sources is vital to gain insight from the powerful abilities KGs lend us to finding patterns in multi-modal data. There is plenty of positivity to take from the ability that ML models must tag types of data in text and suggest a suitable entity to ground terms too if needed. Complexity arises with knowing how much faith to put in a model's ability to do all of this to a high enough standard; with this the requirement of human review is still needed. For this reason, the frameworks mentioned above put high emphasis on modular and transparent benchmarking of specific research tasks [93].

KGs are a very useful interface for data, being both human and machine interpretable. They allow integration of multiple types of data and, subsequently, discovery of patterns not visible to the human eye. Simple relationships between concepts can be weighted and validated in a KG structure allowing thresholds to be used to see above the noise often present in biological data. There are also large benefits to KGs interfacing with LLMs, whether to improve an LLM's reasoning or to be generated by an LLM.

LLMs indeed address longstanding challenges in NLP for drug discovery by processing larger data volumes than traditional ML or manual human efforts. Their expanded context windows, ranging from 1024 tokens in earlier GPT models to 128K in GPT-4 and models like Mixtral, enable handling more comprehensive, multi-step tasks essential for data-intensive fields like drug discovery. Additionally, their advanced natural language understanding (NLU) capabilities allow for immediate applications, for example a prompting task on summarizing lengthy articles on successful drug treatments. This can aid in seeing disease landscapes during early drug development.

LLMs could be valuable in market research, especially when patient-level data is accessible to pharmaceutical companies. They could help uncover the current standard of care for a disease and identify market gaps where a new drug might outperform existing treatments. While the generated outputs still require moderation, LLMs can reduce the manual effort involved by summarizing key insights quickly. This allows researchers to avoid spending days sifting through data, as the model can provide a concise summary of the most relevant outcomes.

The output of LLM pipelines could have severe financial and human health implications if they were to be accepted without verification. While they can accelerate target selection and prioritization in drug discovery, there is a risk of hallucination or the generation of unreliable answers (e.g. concerning target safety). Additionally, LLMs are prone to biases, particularly in areas that are underrepresented in their training data. At present, the pharmaceutical industry does not rely on model outputs as fully accurate without human validation. Scientific experts, or a 'human-in-the-loop' approach, remain essential for fact-checking and supporting the hypotheses generated by models.

Hallucinations, or fabricated outputs, are intrinsic to LLMs and, while they cannot be fully eliminated, they can be reduced. In fields like drug discovery, it is essential for users to understand these limitations and the underlying mechanics. This awareness helps determine when to trust or deploy the models and recognize scenarios where they may fail.

Additionally, there is a trade-off between closed-source and open-source models in terms of control, cost, and flexibility. Closed-source models, accessed through API calls, can reduce the computational overhead, but they come with concerns around data privacy, interpretability, and escalating costs due to frequent usage. Additionally, their lack of transparency makes it difficult to understand how the models function internally. In contrast, open-source models offer greater control and customization, but managing them at scale requires substantial computational resources and technical expertise. A similar trade-off exists when choosing between prompting, fine-tuning, or RAG methods. While prompting is quick and resource-efficient, requiring no infrastructure setup, it may not always provide the depth of understanding that fine-tuning or RAG can offer.

Red teaming – a practice in which a team of experts intentionally tries to find vulnerabilities, weaknesses, or harmful behaviors in the model – is another method used to control the behavior and generation abilities of LLMs, especially to prevent unintended outcomes, such as the exposure of sensitive information like anonymized patient data. However, red teaming is resource-intensive, time-consuming, and prone to expert bias, often failing to cover all edge cases. To address these limitations, integrating red teaming with automated methods within a comprehensive quality assurance pipeline can enhance its effectiveness. For instance, systematically generating adversarial prompts that target ambiguous questions and challenging biomedical terminology, and incorporating counterexamples from benchmark datasets, can improve model robustness testing. Additionally, leveraging biomedical knowledge bases (e.g. UMLS, PubMed, and DrugBank) for automated fact-checking, named entity verification, and consistency checks can complement manual efforts and streamline the overall process.

Biological data are so multifaceted that there is no single perfect way of analysis. Given the complexity of data and research questions asked, it is advised to select data carefully and choose the method of analysis with consideration. Each of the methods covered in this review lends itself to specific use cases; considering the best combination of use cases and available data will lead to the most successful outcome.

Abbreviations

AI	Artificial Intelligence
ATM	Ataxia-Telangiectasia Mutated
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory network
BioBLP	Bio BERT for Link Prediction
BoW	Bag-of-Words
CADD	Computer-Aided Drug Design
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DL	Deep Learning
EFO	Experimental Factor Ontology
EHR	Electronic Health Record
FAIR	Findable Accessible Interoperable Reusable
GloVE	Global Vectors
GPT	Generative Pretrained Transformer
HMM	Hidden Markov Model
KG	Knowledge Graph
LLM	Large Language Model
LSTM	Long Short-Term Memory network
ML	Machine Learning
MeSH	Medical Subject Headings
NER	Named Entity Recognition
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
POS	pPart-Of-Speech
QA	Question Answering
RAG	Retrieval-Augmented Generation
RLHF	Reinforcement Learning from Human Feedback
RNN	Recurrent Neural Network
SGNC	Stochastic Generative Network Complex
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
UID	Unique Identifier
UMLS	Unified Medical Language System

Acknowledgments

In compliance with the International Committee of Medical Journal Editors (ICMJE) recommendations, the authors acknowledge the utilization of AI services in this work: the authors of this article used ChatGPT (GPT-4) by OpenAI exclusively for proofreading for English grammatical and/or language errors. It was also used to help summarise the key messages from the text in the Article Highlights Box. All sentences revised by GPT-4 were reviewed and verified by the authors. No content was generated by the GPT-4 or any other AI service.

Funding

This work is funded by Open Targets, under [OTAR3088] ("Automating Knowledge Management"). B Zdrazil and M Harrison also receive funding from the European Bioinformatics Institute of the European Molecular Biology Laboratory (EMBL-EBI).

Declaration of interest

The authors have no other relevant affiliations or financial involvement with any organization or entity with a financial interest in or financial conflict with the subject matter or materials discussed in the manuscript apart from those disclosed.

Reviewer disclosures

Peer reviewers on this manuscript have no relevant financial or other relationships to disclose.

ORCID

Christine Ann Withers http://orcid.org/0009-0005-5428-7822
Amina Mardiyyah Rufai http://orcid.org/0009-0003-1275-5160
Santosh Tirunagari http://orcid.org/0000-0002-9064-1965
Sebastian Lobentanzer http://orcid.org/0000-0003-3399-6695
Melissa Harrison http://orcid.org/0000-0003-3523-4408
Barbara Zdrazil http://orcid.org/0000-0001-9395-1515

References

1. Sarawagi S. Information extraction. Hanover (MA): Now Publishers Inc; 2008.
2. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Commun ACM. 1975;18(11):613–620. doi: 10.1145/361219.361220
3. Cam LML, Neyman J. Proceedings of the Fifth berkeleysymposium on mathematical statistics and probability. Berkeley and Los Angeles, California: University of California Press; 1967.
4. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. J Mach Learn Res. 2003;3(null):993–1022.
5. Department of Computer Science, Adeyemi College of Education, Ondo, Nigeria, Iroju OG, Olaleke JO. A systematic review of natural language processing in healthcare. IJITCS. 2015;7(8):44–50. doi: 10.5815/ijitcs.2015.08.07
- **This paper is a nice demonstration of the capabilities of NLP in informing drug repositioning theories, showing a real use-case of NLP technologies alongside other drug discovery solutions, leading to a rapid introduction of a drug as a treatment for COVID-19 during the early stages of the pandemic.**
6. Pun FW, Ozerov IV, Zhavoronkov A. Ai-powered therapeutic target discovery. Trends Pharmacol Sci. 2023;44(9):561–572. doi: 10.1016/j.tips.2023.06.010
7. Suryasa IW, Rodríguez-Gámez M, Koldoris T. COVID-19 pandemic. ijhs. 2021;5(2):vi–ix. doi: 10.53730/ijhs.v5n2.2937
8. Smith DP, Oechsle O, Rawling MJ, et al. Expert-augmented computational drug repurposing identified baricitinib as a treatment for COVID-19. Front Pharmacol. 2021;12:709856.
9. Kalil AC, Patterson TF, Mehta AK, et al. Baricitinib plus remdesivir for hospitalized adults with covid-19. N Engl J Med. 2021;384(9):795–807. doi: 10.1056/NEJMoa2031994
10. Wong CH, Siah KW, Lo AW. Estimation of clinical trial success rates and related parameters. Biostatistics. 2019;20(2):273–286. doi: 10.1093/biostatistics/kxx069
11. Kulmanov M, Smaili FZ, Gao X, et al. Semantic similarity and machine learning with ontologies. Brief Bioinform. 2021;22(4):bbaa199. doi: 10.1093/bib/bbaa199
12. Bard JBL, Rhee SY. Ontologies in biology: design, applications and future challenges. Nat Rev Genet. 2004;5(3):213–222. doi: 10.1038/nrg1295
13. Smith B, Ceusters W, Klagges B, et al. Relations in biomedical ontologies. Genome Biol. 2005;6(5):R46. doi: 10.1186/gb-2005-6-5-r46
14. Malone J, Holloway E, Adamusiak T, et al. Modeling sample variables with an experimental factor ontology. Bioinformatics. 2010;26(8):1112–1118. doi: 10.1093/bioinformatics/btq099
15. Wilkinson MD, Dumontier M, Aalbersberg I, et al. The FAIR guiding principles for scientific data management and stewardship. Sci Data. 2016;3(1):160018.
16. Ruas P, Couto FM. NILINKER: attention-based approach to NIL entity linking. J Biomed Inform. 2022;132:104137.
17. Tirunagari S, Saha S, Venkatesan A, et al. Lit-otar framework for extracting biological evidences from literature [Internet]. 2024 [cited 2025 Feb 21]. Available from: http://biorxiv.org/lookup/doi/10.1101/2024.03.06.583722
18. Bornmann L, Haunschild R, Mutz R. Growth rates of modern science: a latent piecewise growth curve approach to model publication numbers from established and new literature databases. Humanit Soc Sci Commun. 2021;8(1):224. doi: 10.1057/s41599-021-00903-w
19. Shilo S, Rossman H, Segal E. Axes of a revolution: challenges and promises of big data in healthcare. Nat Med. 2020;26(1):29–38. doi: 10.1038/s41591-019-0727-5
20. Plake C, Royer L, Winnenburg R, et al. GoGene: gene annotation in the fast lane. Nucleic Acids Res. 2009;37(Web Server):W300–W304. doi: 10.1093/nar/gkp429
21. Ashburner M, Ball CA, Blake JA, et al. Gene ontology: tool for the unification of biology. Nat Genet. 2000;25(1):25–29.
22. Ling MH, Lefevre C, Nicholas KR, et al. Reconstruction of protein-protein interaction pathways by mining subject-verb-objects intermediates. In: Rajapakse J, Schmidt B Volkert G, editors. Pattern recognition in bioinformatics. Berlin Heidelberg: Springer; 2007. p. 286–299.
23. Neo4j graph database & analytics – the Leader in graph databases [internet]. Graph database & analytics. 2025 [cited 2025 Feb 21]. Available from: https://neo4j.com/
24. Lausen G. Relational databases in RDF: keys and foreign keys. In: Christophides V, Collard M Gutierrez C, editors. Semantic web, ontologies and databases [Internet]. Berlin Heidelberg: Springer Berlin Heidelberg; 2008. p. 43–56. [cited 2025 Feb 21]. Available from: https://doi.org/10.1007/978-3-540-70960-2_3
25. Neo4j Aura DB Console. Neo4j. [cited 2025 Apr 1st]. Available from: https://neo4j.com/docs/aura/classic/auradb/getting-started/connect-database/#_neo4j_workspace
26. Crick F. Central dogma of molecular biology. Nature. 1970;227(5258):561–563. doi: 10.1038/227561a0
27. Li L, Wang P, Yan J, et al. Real-world data medical knowledge graph: construction and applications. Artif Intell Med. 2020;103:101817.
28. Zhong L, Wu J, Li Q, et al. A comprehensive survey on automatic knowledge graph construction. ACM Comput Surv. 2024;56(4):1–62. doi: 10.1145/3618295
29. Callahan TJ, Tripodi IJ, Stefanski AL, et al. An open source knowledge graph ecosystem for the life sciences. Sci Data. 2024;11(1):363. doi: 10.1038/s41597-024-03171-w

30. Agrawal R, Prabakaran S. Big data in digital healthcare: lessons learnt and recommendations for general practice. Heredity (Edinb). 2020;124(4):525–534. doi: 10.1038/s41437-020-0303-2
31. Meijer D, Beniddir MA, Coley CW, et al. Empowering natural product science with AI: leveraging multimodal data and knowledge graphs. Nat Prod Rep. 2025. doi: https://doi.org/10.1039/D4NP00008K
32. Daza D, Alivanistos D, Mitra P, et al. BioBLP: a modular framework for learning on multimodal biomedical knowledge graphs. J Biomed Semant. 2023;14(1):20.
33. Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [Internet]. arXiv; 2013 [cited 2025 Feb 21]. Available from: https://arxiv.org/abs/1301.3781
34. Pennington J, Socher R, Manning C. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) [Internet]; Doha, Qatar. Association for Computational Linguistics; 2014. p. 1532–1543. [cited 2025 Feb 21]. Available from: http://aclweb.org/anthology/D14-1162
35. Devlin J, Chang M-W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [Internet]. arXiv. 2018 [cited 2025 Feb 21]. Available from: https://arxiv.org/abs/1810.04805
36. Lee J, Yoon W, Kim S, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020 Feb;36(4):1234–1240. doi: 10.1093/bioinformatics/btz682
37. Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596(7873):583–589.
•• **AlphaFold has been hugely impactful in drug discovery and has immediately become a widely used source of protein structure predictions. The works by DeepMind and EBI have proved invaluable as an open-source database enabling end-users to form hypothesis about potential druggability of proteins.**
38. Shirkhorshidi AS, Aghabozorgi S, Wah TY. A comparison study on similarity and dissimilarity measures in clustering continuous data. Dalby AR, editor. PLOS ONE. 2015;10(12):e0144059.
39. Buniello A, Suveges D, Cruz-Castillo C, et al. Open targets platform: facilitating therapeutic hypotheses building in drug discovery. Nucleic Acids Res. 2025;53(D1):D1467–D1475.
40. Ye C, Swiers R, Bonner S, et al. A knowledge graph-enhanced tensor factorisation model for discovering drug targets. IEEE/ACM Trans Comput Biol Bioinf. 2022;19(6):3070–3080. doi: 10.1109/TCBB.2022.3197320
41. Lamiroy B, Sun T. Computing precision and recall with missing or uncertain ground truth. In: Kwon Y-B Ogier J-M, editors. Graphics recognition new trends and challenges [internet]. Berlin Heidelberg: Springer Berlin Heidelberg; 2013 [cited 2025 Feb 21]. p. 149–162. Available from: https://doi.org/10.1007/978-3-642-36824-0_15
42. Lafferty JD, McCallum A, Pereira FCN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning; San Francisco (CA). Morgan Kaufmann Publishers Inc.; 2001. p. 282–289.
43. Settles B. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. Bioinformatics. 2005;21(14):3191–3192. doi: 10.1093/bioinformatics/bti475
44. Durbin R. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge: Cambridge University Press & Assessment; 1998.
45. Eddy SR. What is a hidden Markov model? Nat Biotechnol. 2004;22(10):1315–1316. doi: 10.1038/nbt1004-1315
46. Zhang H. The optimality of naive Bayes. Aa. 2004;1(2):3.
47. The origins of logistic regression by J.S. Cramer: SSRN [Internet]. [cited 2025 Feb 21]. Available from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=360300
48. Tong S, Koller D. Support vector machine active learning with application sto text classification. In: Proceedings of the Seventeenth International Conference on Machine Learning; San Francisco (CA). Morgan Kaufmann Publishers Inc.; 2000. p. 999–1006.
49. Ye Z, Tafti AP, He KY, et al. SparkText: biomedical text mining on big data framework. PLOS ONE. 2016;11(9):e0162721.
50. Neural Networks From Scratch. VictorZhou.com. 2019 [cited 2025 Apr 4th]. Available from: https://victorzhou.com/series/neural-networks-from-scratch/
51. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–444.
52. Gashaw I, Ellinghaus P, Sommer A, et al. What makes a good drug target?. Drug Discov Today. 2012;17:S24–S30. doi: 10.1016/j.drudis.2011.12.008
53. Katsila T, Spyroulias GA, Patrinos GP, et al. Computational approaches in target identification and drug discovery. Comput Struct Biotechnol J. 2016;14:177–184.
54. Niu Y, Lin P. Advances of computer-aided drug design (CADD) in the development of anti-Azheimer's-disease drugs. Drug Discov Today. 2023;28(8):103665. doi: 10.1016/j.drudis.2023.103665
55. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997;9(8):1735–1780. doi: 10.1162/neco.1997.9.8.1735
56. Choi SR, Lee M. Transformer Architecture and Attention Mechanisms in Genome Data Analysis: A Comprehensive Review. Biology (Basel). 2023;12(7):1033. doi: 10.3390/biology12071033
57. Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need [Internet]. arXiv; 2023 [cited 2024 Sep 12]. Available from: http://arxiv.org/abs/1706.03762
58. Radford A, Narasimhan K, Salimans T, et al. Improving Language Understanding by Generative Pre-Training.
59. Fang L, Chen Q, Wei C-H, et al. Bioformer: an efficient transformer language model for biomedical text mining [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: https://arxiv.org/abs/2302.01588
60. Beltagy I, Lo K, Cohan A. SciBERT: A Pretrained Language Model for Scientific Text [Internet]. arXiv; 2019 [cited 2025 Apr 2]. Available from: http://arxiv.org/abs/1903.10676
61. Gu Y, Tinn R, Cheng H, et al. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. ACM Trans Comput Healthcare. 2022;3(1):1–23. doi: 10.1145/3458754
62. Zhou Y, Srikumar V. A closer look at how fine-tuning changes BERT [Internet]. arXiv; 2021 [cited 2025 Feb 21]. Available from: https://arxiv.org/abs/2106.14282
63. Razuvayevskaya O, Lopez I, Dunham I, et al. Genetic factors associated with reasons for clinical trial stoppage. Nat Genet. 2024;56(9):1862–1867. doi: 10.1038/s41588-024-01854-z
64. Minikel EV, Painter JL, Dong CC, et al. Refining the impact of genetic evidence on clinical success. Nature. 2024;629(8012):624–629.
65. Kühnel L, Fluck J. We are not ready yet: limitations of state-of-the-art disease named entity recognizers. J Biomed Semantics. 2022;13(1):26.
66. Pakhale K. Comprehensive overview of named entity recognition: models, domain-specific applications and challenges [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: https://arxiv.org/abs/2309.14084
67. Jehangir B, Radhakrishnan S, Agarwal R. A survey on Named Entity Recognition — datasets, tools, and methodologies. Nat Language Process J. 2023;3:100017. doi: 10.1016/j.nlp.2023.100017
68. Beltagy I, Peters ME, Cohan A. Longformer: the long-document transformer [Internet]. arXiv; 2020 [cited 2025 Feb 21]. Available from: https://arxiv.org/abs/2004.05150
69. Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback [Internet]. arXiv; 2022 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2203.02155
70. Wu J. Literature review on vulnerability detection using NLP technology [Internet]. arXiv. 2021 [cited 2025 Feb 21]. Available from: https://arxiv.org/abs/2104.11230
71. Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners. 2019 [cited 2025 Feb 21]. Available from: https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe

72. Brown T, Mann B, Ryder N, et al. Language models are few-shot learners. Advances in Neural Information Processing Systems [Internet]. Curran Associates, Inc.; 2020 [cited 2025 Feb 21]. p. 1877–1901. Available from: https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html
73. GPT-4 Technical Report [Internet]. [cited 2025 Feb 21]. Available from: https://arxiv.org/html/2303.08774v6
74. Introducing the next generation of Claude [Internet]. [cited 2025 Feb 21]. Available from: https://www.anthropic.com/news/claude-3-family
75. Team G, Anil R, Borgeaud S, et al. Gemini: a family of highly capable multimodal models [Internet]. arXiv. 2023 [cited 2025 Feb 21]. Available from: https://arxiv.org/abs/2312.11805
76. Team G, Georgiev P, Lei VI, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context [Internet]. arXiv; 2024 [cited 2025 Feb 21]. Available from: https://arxiv.org/abs/2403.05530
77. Touvron H, Lavril T, Izacard G, et al. LLaMA: open and efficient foundation language models [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2302.13971
78. Touvron H, Martin L, Stone K, et al. Llama 2: open foundation and fine-tuned chat models [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2307.09288
79. Grattafiori A, Dubey A, Jauhri A, et al. The llama 3 herd of models [Internet]. arXiv. 2024 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2407.21783
80. Jiang AQ, Sablayrolles A, Mensch A, et al. Mistral 7B [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2310.06825
81. Jiang AQ, Sablayrolles A, Roux A, et al. Mixtral of experts [Internet]. arXiv; 2024 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2401.04088
82. Almazrouei E, Alobeidli H, Alshamsi A, et al. The falcon series of open language models [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2311.16867
83. Abdin M, Aneja J, Awadalla H, et al. Phi-3 technical report: a highly capable language model locally on your phone [Internet]. arXiv; 2024 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2404.14219
84. Anil R, Dai AM, Firat O, et al. PaLM 2 technical report [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2305.10403
85. Hugging face – The AI community building the future. Internet. 2025 [cited 2025 Feb 21]. Available from: https://huggingface.co/
86. Luo R, Sun L, Xia Y, et al. BioGPT: generative pre-trained transformer for biomedical text generation and mining. Brief Bioinform. 2022;23(6):bbac409. doi: 10.1093/bib/bbac409
87. Luo Y, Zhang J, Fan S, et al. BioMedGPT: open multimodal generative pre-trained transformer for BioMedicine [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2308.09442

• **This paper introduces a multimodal LLM for biomedical applications "BioMEDGPT". It extends traditional LLMs by incorporating multimodal capabilities, making it a crucial resource for biomedical NLP. This is particularly valuable for drug discovery, where insights often emerge from combining diverse modalities, such as molecular interactions, clinical literature, and genomic data.**

88. Luo L, Ning J, Zhao Y, et al. Taiyi: a bilingual fine-tuned large language model for diverse biomedical tasks. J Am Med Inform Assoc. 2024;31(9):1865–1874.
89. Wang R, Feng H, Wei G-W. ChatGPT in drug discovery: a case study on anti-cocaine addiction drug development with chatbots [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2308.06920
90. Huang L, Yu W, Ma W, et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ACM Trans Inf Syst. 2025;43(2):1–55. doi: 10.1145/3703155
91. Zhu Y, Yuan H, Wang S, et al. Large language models for information retrieval: a survey [Internet]. arXiv; 2024 [cited 2025 Apr 2]. Available from: http://arxiv.org/abs/2308.07107
92. Wu K, Wu E, Zou J. ClashEval: quantifying the tug-of-war between an LLM's internal prior and external evidence [Internet]. arXiv; 2025 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2404.10198
93. Lobentanzer S, Feng S, Bruderer N, et al. A platform for the biomedical application of large language models. Nat Biotechnol. 2025;43(2):166–169.
94. Pan S, Luo L, Wang Y, et al. Unifying large language models and knowledge graphs: a roadmap [Internet]. 2024 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2306.08302
95. Dagdelen J, Dunn A, Lee S, et al. Structured information extraction from scientific text with large language models. Nat Commun. 2024;15(1):1418.
96. Wei J, Tay Y, Bommasani R, et al. Emergent Abilities of large language models [Internet]. arXiv; 2022 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2206.07682
97. Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2201.11903
98. Yao S, Yu D, Zhao J, et al. Tree of thoughts: deliberate problem solving with large language models [Internet]. arXiv; 2023 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2305.10601
99. Wornow M, Lozano A, Dash D, et al. Zero-shot clinical trial patient matching with LLMs [Internet]. arXiv; 2024 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2402.05125
100. Chaves JMZ, Wang E, Tu T, et al. Tx-LLM: a large language model for therapeutics [Internet]. arXiv; 2024 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2406.06316
101. Stanford CRFM [Internet]. [cited 2025 Feb 21]. Available from: https://crfm.stanford.edu/2023/03/13/alpaca.html
102. Han W, Fang M, Zhang Z, et al. MedINST: Meta Dataset of Biomedical Instructions. In: Al-Onaizan Y, Bansal M Chen Y-N, editors. Findings of the Association for Computational Linguistics: EMNLP 2024 [Internet]. Miami (FL): Association for Computational Linguistics; 2024 [cited 2025 Apr 2]. p. 8221–8240. Available from: https://aclanthology.org/2024.findings-emnlp.482/
103. Tran H, Yang Z, Yao Z, et al. BioInstruct: instruction tuning of large language models for biomedical natural language processing. J Am Med Inform Assoc. 2024;31(9):1821–1832. doi: 10.1093/jamia/ocae122

•• **This work is key to advancing instruction tuning for biomedical NLP. It demonstrates the impact of instruction tuning on improving biomedical text comprehension. As the field evolves this is crucial for building models that better understand domain-specific queries for downstream applications.**

104. Zhang X, Tian C, Yang X, et al. AlpaCare: instruction-tuned large language models for medical application [Internet]. arXiv; 2025 [cited 2025 Apr 2]. Available from: http://arxiv.org/abs/2310.14558
105. Clark MM, Hildreth A, Batalov S, et al. Diagnosis of genetic diseases in seriously ill children by rapid whole-genome sequencing and automated phenotyping and interpretation. Sci Transl Med. 2019;11(489):eaat6177.
106. Vora B, Kuruvilla D, Kim C, et al. Applying Natural Language Processing to ClinicalTrials.gov: mRNA cancer vaccine case study. Clin Transl Sci. 2023;16(12):2417–2420.
107. Lee J-S, Hsiang J. Patent classification by fine-tuning BERT language model. World Patent Inf. 2020;61:101965.
108. Tyagi T, Magdamo CG, Noori A, et al. NeuraHealth: an automated screening pipeline to detect undiagnosed cognitive impairment in electronic health records with deep learning and natural language processing [Internet]. arXiv; 2022 [cited 2025 Feb 21]. Available from: http://arxiv.org/abs/2202.00478
109. Lobentanzer S, Aloy P, Baumbach J, et al. Democratizing knowledge representation with BioCypher. Nat Biotechnol. 2023;41(8):1056–1059.
110. Hulcoop DG, Trynka G, McDonagh EM. Open Targets: 10 years of partnership in target discovery. Nat Rev Drug Discov [Internet]. 2024 [cited 2025 Feb 21]. doi: 10.1038/d41573-024-00204-2