Related papers:

BioRED: A Rich Biomedical Relation Extraction Dataset -
https://arxiv.org/ftp/arxiv/papers/2204/2204.04263.pdf

Comparison of biomedical relationship extraction methods and models for knowledge graph creation
https://arxiv.org/pdf/2201.01647.pdf
BioREx: Improving Biomedical Relation Extraction by Leveraging Heterogeneous Datasets-
https://arxiv.org/pdf/2306.11189.pdf

SCI BERT: A Pretrained Language Model for Scientific Text
https://arxiv.org/pdf/1903.10676.pdf

A Data-driven Approach for Noise Reduction in Distantly Supervised Biomedical Relation Extraction
https://arxiv.org/pdf/2005.12565.pdf

Deep Bidirectional Transformers for Relation Extraction without Supervision
https://arxiv.org/pdf/1911.00313.pdf

Relation Extraction using Multiple Pre-Training Models in the Biomedical Domain
https://aclanthology.org/2021.ranlp-1.60.pdf

A Hybrid Model with Pre-trained Entity-Aware Transformer for Relation Extraction
https://www.semanticscholar.org/paper/ef2ceaad8242d14b85a544ec312f1f24da05def4

BioREx: Improving biomedical relation extraction by leveraging heterogeneous datasets
https://pubmed.ncbi.nlm.nih.gov/37673376/

Connected graph -

1. https://www.connectedpapers.com/main/175dcaf0e3a8a6c702e13f1d1d656ff31484b66a/BioRED%3A-a-rich-biomedical-relation-extraction-dataset/graph
2. https://www.connectedpapers.com/main/6d396877819c09403a63f3c1fe4fb47513f6a5f8/A-Data%20driven-Approach-for-Noise-Reduction-in-Distantly-Supervised-Biomedical-Relation-Extraction/graph

BioRED: A Rich Biomedical Relation Extraction Dataset

600 - PubMed abstracts used for RE only

BioBERT: a pre-trained biomedical language representation model for biomedical text mining

Abstract: The abstract introduces BioBERT as a specialized pre-trained language model for the biomedical domain, trained on PubMed and PMC datasets. It highlights its significance in biomedical text mining and its potential to improve various NLP tasks.

Introduction: The introduction emphasizes the importance of biomedical text mining and the limitations of using general-purpose language models in this domain. It introduces BioBERT as a solution to address these challenges and improve the understanding of biomedical text.

Background: This section provides context on pre-trained language models like BERT and the need for domain-specific models in biomedicine. It explains the data sources and training process for BioBERT, emphasizing its ability to capture biomedical terminology.
Methods: The methods section details the architecture and training procedure of BioBERT, including tokenization, fine-tuning, and model evaluation. It also mentions the utilization of domain-specific data and vocabulary to enhance model performance.

Experiments: The experiments section presents the evaluation results of BioBERT on various biomedical NLP tasks, demonstrating its effectiveness in tasks such as named entity recognition, relation extraction, and question answering. It compares BioBERT's performance with other models.

Discussion: This section discusses the strengths and limitations of BioBERT, highlighting its versatility in biomedical text-mining tasks. It also suggests potential future improvements and applications.

Conclusion: The conclusion summarizes the significance of BioBERT in biomedical NLP and its potential impact on research and applications in the field.

Pros of BioBERT:

- Specialized in the biomedical domain, capturing domain-specific terminology.
- Improved performance in various biomedical NLP tasks.
- Enhanced understanding of biomedical literature and data.

Cons of BioBERT:

- Requires domain-specific pre-training data, which may limit its applicability in other domains.
- May not capture highly specialized or rare biomedical terms.
- Performance can be influenced by the quality and quantity of the training data.

Comparison of biomedical relationship extraction methods and models for knowledge graph creation

- Relationship extraction is the process of identifying and classifying relationships between named entities in a text.
- The authors generated a knowledge graph with links to the evidence sentences, based on the extracted and normalized relationships from PubMed.
- **datasets** for supervised biomedical relationship extraction -BioCreative VI and VII ChemProt datasets for chemical protein interactions, BioCreative V Chemical-disease (CDR) dataset, ADE Corpus, BioInfer, DDI'13, n2c2 2018 ADE, N-ary, and BioRED. (drug-drug or protein-protein interactions are out of the scope of this study)

Following relationship classes:

- Drug-Gene relationships - Up-regulator/activator, Down-regulator/inhibitor, Regulator, Part of, Modulator, Co-factor, Substrate or product of
- Drug-Disease relationships - Therapeutic Use/Treatment, Cause/Adverse event, Decrease Disease, Increase Disease, Effect on, Biomarker
- Gene-Disease relationship -

Methods for comparison

1. **Rule-based**
2. **ML-based -** Naive Bayes, RF,
3. **DL -** DistilBERT, text to text T5, PubMedBERT, SciFive

Knowledge graphs (knowledge is represented as graphs of interrelated concepts [ref]) benefits of KG (semantic information). applications of KG in QA products (such as alexa, siri, google assistant. similarly, KG can bring accelarating drug discovery, development, indication expansion of existing drugs, and pharmacovigilance. In order to extract information and structure them for entry into the knowledge graph, it is necessary to perform named entity recognition of relevant entities (for bio-medicine these could be genes/targets, compounds, diseases, cell lines, pathways, organs, etc.), One of the first systems to attempt relationship extraction in the biomedical domain was EDGAR (extraction relations b.w drugs and genes in cancer domain.) several approaches to structuring relationships were explored: Existence of relationship between entities - classifies whether there is an actual semantic relationship between two entities Extracting predicate verb as relationship type - Normalizing relationship types Method - NER, relationship data model (drug gener, drug disease, gene disease, Together with the relationship classes, if available, mode of action is an important modifier for Gene-Disease relationships.

SCI BERT: A Pretrained Language Model for Scientific Text

The paper presents SCIBERT, a language model built upon BERT, specifically tailored for scientific domain natural language processing (NLP) tasks. The motivation arises from the scarcity and high cost of labelled scientific data for training NLP models. SCIBERT is pre trained using a large-scale corpus of scientific publications, aiming to enhance performance on various downstream scientific NLP tasks, such as sequence tagging, sentence classification, and dependency parsing.

Introduction:

- **Challenges in Scientific NLP**: Obtaining labelled scientific data for NLP tasks is expensive due to the expertise required for annotation.
- **Motivation for SCIBERT**: SCIBERT is introduced to mitigate this challenge by leveraging unsupervised pre-training on a substantial corpus of scientific texts, providing a resource for scientific NLP tasks.

Methods:

- **Model Architecture**: SCIBERT follows BERT's architecture but is pretrained on scientific text instead of general domain data.
- **Vocabulary**: SCIBERT employs SCIVOCAB, a specialised vocabulary created from scientific texts, as opposed to BASEVOCAB used by BERT.
- **Training Corpus**: SCIBERT is trained on 1.14M full text papers from various scientific domains, resulting in a 3.17B token corpus.
- **Experimental Setup**: The paper conducts experiments on tasks such as Named Entity Recognition (NER), PICO extraction, Text Classification, Relation Classification, and Dependency Parsing using different datasets from biomedical, computer science, and multidomain domains.

Results and Discussion:

- **Performance Comparison**: SCIBERT significantly outperforms BERT-Base on scientific tasks, achieving a +2.11 F1 improvement with finetuning and a +2.43 F1 improvement without fine tuning. In addition, SCIBERT achieves new state-of-the-art results on several scientific NLP tasks, including biomedical tasks such as BC5CDR and ChemProt, and EBM-NLP.
- **Effect of Fine Tuning**: Finetuning SCIBERT yields better results than task-specific architectures atop frozen embeddings, especially in the computer science and biomedical domains (+3.25 F1 with SCIBERT and +3.58 with BERT-Base, on average).
- **Importance of SCIVOCAB**: SCIBERT's performance benefits from the in-domain vocabulary (SCIVOCAB), showing improvements across all scientific domains.

Related Work:

- **Comparison with Other Models**: SCIBERT is contrasted with similar models like BIOBERT and CLINICALBERT, highlighting its unique approach to training on a diverse scientific corpus.

Conclusion and Future Work:

- **SCIBERT's Achievements**: SCIBERT demonstrates superiority over BERT-Base and even outperforms some reported BIOBERT results in biomedical tasks.
- **Future Directions**: The authors plan to expand SCIBERT into a larger resource, akin to BERT-Large, and explore training strategies considering different proportions of papers from various scientific domains.

Pros:

- The paper introduces a new pretrained language model, SCI-BERT, that is specifically designed for scientific text and outperforms other models on several downstream NLP tasks.
- It has experiments on several new datasets, including SciERC and ACL-ARC, which could be useful for future research in the scientific NLP domain.

Cons:

- There is no detailed analysis of the limitations of SCI-BERT or potential failure cases.
- The paper does not include an ablation study to determine the relative importance of different components of the SCI-BERT architecture.
- It does not provide a detailed analysis of the computational requirements or training time for SCI-BERT, which could be important for practical applications.

Deep Bidirectional Transformers for Relation Extraction without Supervision

This paper presents a novel framework for relation extraction without supervision, which leverages syntactic parsing and pre-trained word embeddings to extract precise relations. The method employs a small set of precise relations to annotate a larger corpus, which is then used to fine-tune a pre-trained BERT model for relation extraction. The proposed method significantly outperforms two unsupervised baselines and achieves state-of-the-art results in three out of four biomedical data sets. The paper also discusses related work in relation extraction and the limitations of the proposed method.

This paper is unique in its approach to relation extraction without supervision. Unlike previous works that rely on gold data or distant supervision, this paper proposes a novel framework that leverages syntactic parsing and pre-trained word embeddings to extract precise relations, which are then used to annotate a larger corpus for fine-tuning a pre-trained BERT model. The proposed method is able to successfully fine-tune a large pre-trained language model with noisy data, which is a departure from previous works that rely on gold data for fine-tuning. The paper also presents empirical evaluation results on four biomedical data sets, which show that the proposed method significantly outperforms two unsupervised baselines and achieves state-of-the-art results in three out of four cases. This

is a significant improvement over previous works that rely on distant supervision or gold data.

1. **Data Sets and Setup**: The paper evaluates the proposed method on four biomedical data sets expressing disease-drug and disease-gene relations. Three of these data sets are well-known benchmark data sets for relation extraction, while the fourth is a proprietary manually curated data set. The paper considers only sentence-level relations and splits the CDR instances into sentences.

2. **Baselines**: The paper compares the proposed method against two unsupervised baselines: a BERT model trained with gold or distantly supervised data and the current state-of-the-art.

3. **Evaluation Metric**s: The paper uses standard evaluation metrics for relation extraction, including precision, recall, and F1 score.

4. **Results**: The paper presents the results of the experiments, showing that the proposed method significantly outperforms the two unsupervised baselines and achieves state-of-the-art results in three out of four biomedical data sets. The paper also provides a detailed analysis of the results, including error analysis and ablation studies.

5. Pros: - The proposed method is able to extract precise relations without the need for supervision, which is a departure from previous works that rely on gold data or distant supervision.

- The method leverages syntactic parsing and pre-trained word embeddings to annotate a larger corpus for fine-tuning a pre-trained BERT model, which allows for the successful fine-tuning of a large pre-trained language model with noisy data.

- The proposed method achieves state-of-the-art results in three out of four biomedical data sets, which is a significant improvement over previous works that rely on distant supervision or gold data.

6. Cons: - The proposed method is limited to sentence-level relations, which may not capture more complex relations that span multiple sentences or paragraphs.

- The method relies on syntactic parsing, which may introduce errors and limit the generalizability of the method to other domains or languages.

- The method requires a small set of precise relations to annotate a larger corpus, which may be difficult to obtain in some domains or languages.

A Data-driven Approach for Noise Reduction in Distantly Supervised Biomedical Relation Extraction

- Obtaining fine grained relations for the biomedical domain is challenging due to not only the annotation costs, but the added requirement of domain expertise.
- Distant supervision (DS) provides a meaningful way to obtain large-scale data for RE but this form of data collection also tends to result in an increased amount of noise.
- We extend sentence-level relation enriched BERT to bag-level MIL.
- The goal of distantly supervised bag-level MIL for corpus-level RE is then to predict the missing relation ‘r’ given the bag.
- Using special markers for entities with BERT in the order they appear in a sentence encodes the positional information that improves the performance of sentence-level RE.
- In contrast, for bag-level distant supervision, the noisy channel can be attributed to several factors for a given triple (h, r, t) and bag Bg.

- This paper addresses the problem of relation extraction (RE) in the biomedical domain, where the amount of unstructured scientific texts continues to grow, making manual annotation of these texts for the task of RE increasingly expensive. Distant supervision offers a viable approach to combat this by quickly producing large amounts of labeled, but considerably noisy, data. The authors aim to reduce such noise by extending an entity-enriched relation classification BERT model to the problem of multiple instance learning (MIL), and defining a simple data encoding scheme that significantly reduces noise, reaching state-of-the-art performance for distantly-supervised biomedical relation extraction.

- The authors first introduce the problem of RE and the challenges associated with it, including the need for large amounts of labelled data, the difficulty of capturing complex relations, and the presence of noisy data. They then describe the distant supervision approach, which involves using existing knowledge graphs (KGs) to automatically label text data. However, this approach suffers from the problem of noisy labelling, where not all sentences containing entity pairs are actually expressing the relation of interest.

- To address this problem, the authors propose a MIL-based approach that considers bags of sentences containing entity pairs, rather than individual sentences. They extend the entity-enriched relation classification BERT model for MIL, and introduce a k-tag encoding scheme that encodes knowledge about the direction of relation triples. The k-tag encoding scheme involves tagging each sentence in a bag with a k-tag that indicates the position of the entity pair in the sentence, and using this information to train the model to distinguish between different relation types.

- **Sentence ordered**: Called s-tag, entities are marked in the order they appear in the sentence.
- **KB ordered**: Called k-tag, entities are marked in the order they appear in the KB.

- The authors evaluate their approach on two biomedical relation extraction datasets, and show that it outperforms existing methods. They also conduct ablation studies to demonstrate the effectiveness of their approach and analyze the impact of different components. They find that the k-tag encoding scheme significantly reduces noise and improves performance, and that the entity-enriched relation classification BERT model is effective for MIL-based RE. They also analyze the impact of different hyperparameters and show that their approach is robust to changes in these parameters.

- Overall, this work highlights the importance of data quality in distant supervision and provides a simple yet effective methodology for reducing noise in biomedical relation extraction. The authors demonstrate that their approach achieves state-of-the-art performance on two datasets, and provide insights into the impact of different components and hyperparameters.

BioREx: Improving biomedical relation extraction by leveraging heterogeneous dataset .Paper Link

Tasks

List of Datasets for BioMedical Relation Extraction

- BioRED Dataset: The BioRED dataset is a large-scale gene-disease association dataset for biomedical relation extraction. It contains information related to gene-disease associations and is available on Zenodo. This dataset is useful for tasks involving the extraction of gene-disease relationships from biomedical text. [2]
- AImed Dataset: The AImed dataset is designed for protein name recognition and protein-protein interaction (PPI) extraction. It consists of 750 Medline abstracts and is used to develop and evaluate these specific aspects of biomedical relation extraction. [3]
- **BioRel:** BioRel is a dataset used in biomedical relation extraction tasks. It includes subtasks like Bacteria Biotope and Seed Development subtasks, making it suitable for various biomedical relation extraction experiments. [5]
- Chemrpot (**link**)

- DDI (link) - The DDIExtraction 2013 task relies on the DDI corpus which contains MedLine abstracts on drug-drug interactions as well as documents describing drug-drug interactions from the DrugBank database.
- I2b2
- **GAD** (**link**) - GAD, or Gene Associations Database, is a corpus of gene-disease associations curated from genetic association studies.
- CMeIE (link) -Chinese Medical Information Extraction, a dataset that is also released in CHIP2020, is used for CMeIE task. The task is aimed at identifying both entities and relations in a sentence following the schema constraints. There are 53 relations defined in the dataset, including 10 synonymous sub-relationships and 43 other sub-relationships.
- EU-ADR (Link) - The EU-ADR corpus is a biomedical relation extraction dataset that contains 100 abstracts, with relations between drugs, disorder, and targets.
- BLUE - The BLUE benchmark consists of five different biomedicine text-mining tasks with ten corpora. These tasks cover a diverse range of text genres (biomedical literature and clinical notes), dataset sizes, and degrees of difficulty and, more importantly, highlight common biomedicine text-mining challenges.

Corpus	Train	Dev	Test	Task	Metrics	Domain	Avg sent len
MedSTS, sentence pairs	675	75	318	Sentence similarity	Pearson	Clinical	25.8
BIOSSES, sentence pairs	64	16	20	Sentence similarity	Pearson	Biomedical	22.9
BC5CDR-disease, mentions	4182	4244	4424	NER	F1	Biomedical	22.3
BC5CDR-chemical, mentions	5203	5347	5385	NER	F1	Biomedical	22.3
ShARe/CLEFE, mentions	4628	1075	5195	NER	F1	Clinical	10.6
DDI, relations	2937	1004	979	Relation extraction	micro F1	Biomedical	41.7
ChemProt, relations	4154	2416	3458	Relation extraction	micro F1	Biomedical	34.3
i2b2 2010, relations	3110	11	6293	Relation extraction	F1	Clinical	24.8
HoC, documents	1108	157	315	Document classification	F1	Biomedical	25.3
MedNLI, pairs	11232	1395	1422	Inference	accuracy	Clinical	11.9

Table 1: BLUE tasks

- **BC5CDR**: A dataset for chemical-disease relation extraction from biomedical literature, consisting of 1,500 PubMed abstracts. BC5CDR Dataset
- **JNLPBA**: A dataset for named entity recognition in the biomedical domain, consisting of 2,000 PubMed abstracts. JNLPBA Dataset
- **NCBI-disease**: A dataset for disease named entity recognition, consisting of 793 PubMed abstracts. NCBI-disease Dataset
- **EBM-NLP**: A dataset for PICO element extraction in clinical trial abstracts, consisting of 1,000 abstracts. EBM-NLP Dataset
- **GENIA**: A semantically annotated corpus for bio-text mining, consisting of 2,000 MEDLINE abstracts. GENIA Dataset
- **ChemProt**: A dataset for chemical-protein relation extraction, consisting of 1,000 PubMed abstracts. ChemProt Dataset

- **SciERC**: A dataset for entity and relation classification in computer science abstracts, consisting of 500 abstracts. SciERC Dataset
- **ACL-ARC**: A dataset for intent classification in scientific papers that cite other papers, consisting of 5,000 sentences. ACL-ARC Dataset
- **Paper Field**: A dataset for field classification of scientific papers, consisting of 7 fields and 12,000 training examples. Paper Field Dataset
- **SciCite**: A dataset for citation intent classification in scientific papers, consisting of 27,000 sentences. SciCite Dataset
- **ChemDisGene** -: a collection of Biomedical research abstracts annotated with mentions of Chemical, Disease, and Gene/Gene-product entities, and pairwise relationships between those entities. CZI Science is releasing this data to promote NLP research on Relation Extraction from Biomedical text.Dataset github

Dataset links-

https://huggingface.co/bigbio

Chemprot dataset- (Dataset -description - link)

- Understand the dataset -
    - ChemProt consists of 1,820 PubMed abstracts with chemical-protein interactions and was used in the BioCreative VI text mining chemical-protein interactions shared task We use the standard training and test sets in the ChemProt shared task and evaluate the same five classes: CPR:3, CPR:4, CPR:5, CPR:6, and CPR:9.
- relations are directed, always connecting a GENE-type entity (gene or protein) to a CHEMICAL-type entity.
    - Contains abstracts, entities, and relation types in 3 separate files.
    - Combine 3 files based on PMID
    - We perform NLP preprocessing steps on the Title and Abstract columns
        - Use the TEES system (link) to run a preprocessing pipeline of tokenization, POS tagging, and parsing
    - Start_chr_offset and end_chr_offset give us the exact location of the entity in the abstract

Relation.tsv

	PMID	CPR	Evaluation_type	CPR	Arg1	Arg2
0	10047461	CPR:3	Y	ACTIVATOR	Arg1:T13	Arg2:T57
1	10047461	CPR:3	Y	ACTIVATOR	Arg1:T7	Arg2:T39

Entity.tsv

	PMID	Entity_No	Entity_type	start_chr_offset	end_chr_offset	text_str_entity
0	23538162	T1	CHEMICAL	1305	1308	Rg1
1	23538162	T2	CHEMICAL	291	306	Ginsenoside Rg1

Abstract.tsv

	PMID	Title	Abstract
0	10471277	Probing the salmeterol binding site on the bet...	Salmeterol is a long-acting beta2-adrenergic r...
1	23150485	Induction of multidrug resistance transporter ...	The multidrug transporter, breast cancer resis...

How to interpret chemprot dataset

Text Data: The text data is divided into two main categories:

- "title and abstract" with the ID "1" contains a detailed description of a research article.
- "CHEMICAL" and "GENE" with various IDs correspond to entities mentioned in the text.

Entity Types:

- "CHEMICAL" entities represent chemical compounds and are mentioned in the text.
- "GENE-N" entities represent gene names mentioned in the text.
- "GENE-Y" entities represent entities related to genes, which could be genes themselves or other gene-related entities.

Entity Offsets: The "offsets" indicate the character positions of the entities within the text data. For example, "Salmeterol" starts at character position 135 and ends at 145.
Relations: The data also contains information about relations between entities. Each relation is annotated with a "type," which describes the nature of the relation between two entities. For example, "Agonist" indicates that the first entity (arg1) is an agonist of the second entity (arg2).

Feature Description

- PMID: The PubMed identifier of the text passage from which the entity or relationship was extracted.
- Title: The title of the text passage from which the entity or relationship was extracted.
- Abstract: The abstract of the text passage from which the entity or relationship was extracted.
- Entity_No: A unique identifier for the entity or relationship.
- Entity_type: The type of entity or relationship.
- start_chr_offset: The start character offset of the entity or relationship in the text passage.
- end_chr_offset: The end character offset of the entity or relationship in the text passage.
- text_str_entity: The text of the entity or relationship.
- CPR_group: A unique identifier for the chemical-protein-relation (CPR) group.
- Evaluation_type: The type of evaluation used to annotate the entity or relationship.
- CPR_relation: The type of CPR relationship.
- Arg1: The first argument of the CPR relationship.
- Arg2: The second argument of the CPR relationship.

The example Chemprot dataset that describes the following CPR relationship:
CPR_relation: DIRECT-REGULATOR
Arg1: cigarette smoke (T7)
Arg2: Nrf2 (T21)

This means that cigarette smoke directly regulates Nrf2.

The CPR_relation column can have the following values:

- DIRECT-REGULATOR: The first argument directly regulates the second argument.
- INDIRECT-REGULATOR: The first argument indirectly regulates the second argument.
- ACTIVATOR: The first argument activates the second argument.
- INHIBITOR: The first argument inhibits the second argument.
- SUBSTRATE: The first argument is a substrate of the second argument.
- PRODUCT: The first argument is a product of the second argument.
- COFACTOR: The first argument is a cofactor of the second argument.
- BINDING: The first argument binds to the second argument.

GAD-

Here is an example of a sentence from the GAD dataset, along with its label:

Sentence: Mutations in the BRCA1 gene can cause breast cancer.
Label: 1 (positive)
Sentence: The BRCA1 gene is not associated with lung cancer.
Label: 0 (negative)

BioRel:

The BioRel dataset comprises 124 labels representing actual relations and a "NA" (Not A relation) label indicating the absence of a relation between two entities. Covering a wide range of relations in the biomedical domain, including treatment, component of, side effect, and metabolic mechanism, this dataset is notable for the significant presence of "NA" instances, which reflects the sparsity of relations between entities in real biomedical scenarios. This highlights the dataset's realistic representation of the likelihood of entities having no relevant relation. Additionally, BioRel introduces challenges for relation extraction models by considering attributes such as symmetric and asymmetric relations, exemplified by including both "may treat" and "may be treated by" relations.

Example of a relation:

"sentence": "the peculiar distribution and partition on fracture of the envelope glycoproteins on the inner nuclear membrane are similar to those of sindbis virus envelope glycoproteins on the plasma membrane of infected cells .",
"relation": "**anatomic_structure_is_physical_part_of**",
"lexical_feature0": "the|plasma|membrane|of|infected|cells|.",
"lexical_feature1": "on|the|plasma|membrane|of|infected|cells|.|PAD",
"lexical_feature2":"glycoproteins|on|the|plasma|membrane|of|infected|cells|.|PAD|PAD",
"syntactic_feature0": "plasma|2|compound",
"syntactic_feature1": "membrane|0|root",
"syntactic_feature2": "of|5|case",
"syntactic_feature3": "infected|5|amod",
"syntactic_feature4": "cells|2|nmod"
}

Example of a reaction with NA:

"sentence": "the 14c label distribution in the isolated ribose precluded a simple hexose-to-pentose conversion by elimination of one terminal carbon from mannitol-1-phosphate .",
"relation": "**NA**",
"lexical_feature0": "isolated|ribose|precluded|a|simple|hexose-to-pentose|conversion",
"lexical_feature1": "the|isolated|ribose|precluded|a|simple|hexose-to-pentose|conversion|by",
"lexical_feature2": "in|the|isolated|ribose|precluded|a|simple|hexose-to-pentose|conversion|by|elimination",
"syntactic_feature0": "ribose|2|nsubj",
"syntactic_feature1": "precluded|0|root",
"syntactic_feature2": "a|5|det",
"syntactic_feature3": "simple|5|amod",
"syntactic_feature4": "hexose|2|obj",
"syntactic_feature5": "-|5|punct",
"syntactic_feature6": "to|9|case",
"syntactic_feature7": "-|9|punct",

"syntactic_feature8": "pentose|5|nmod"
},

Understanding the dataset:
The training set of the dataset consists of 534,406 sentences, which have been divided into 39,969 bags. In the validation set, there are 218,669 sentences distributed across 15,892 bags. Lastly, the testing set includes 114,515 sentences organized into 20,759 bags. Each bag in the dataset is defined by containing sentences that share the same head and tail entities. On average, there are approximately 13 sentences per bag in the training set, 5 sentences per bag in the validation set, and 8 sentences per bag in the testing set.

Beyond the training, test, and dev sets, the BioRel dataset also provides us with word embeddings, which according to them, are adopted from BioWordVec. Finally, there is a file which allows us to label the relationships.

EU-ADR(European Union Adverse Drug Reaction)

Examples -

1. Target-Drug True concept Caspase-3 0 9 annotator1,Computer,annotator2,annotator3 ['uniprot/3037912'] 0 Genes & Molecular Sequences
2. Target-Drug True concept manganese 65 74 annotator1,annotator2,annotator3 ['sda/2', 'sda/1'] 0 Chemicals & Drugs

   Columns:
   - Target-Drug: Indicates the type or category of the annotation (e.g., concept, relation).
   - True/False: Represents whether the annotation is considered true or false.
   - Concept: Specifies the type of concept being annotated (e.g., protein names, chemicals).
   - Start and End Positions: Denote the character positions of the annotated text within the larger document.
   - Annotators: Lists the individuals or systems responsible for making the annotations.
   - IDs: Unique identifiers associated with the annotated entities.
   - Cluster IDs: Group identifiers for coreferent mentions.
   - Section: Indicates the section or category of the annotated data (e.g., Genes & Molecular Sequences, Chemicals & Drugs).

   Example Interpretation:
   - For instance, in the first row, the annotation is related to the concept "Caspase-3," starting at position 0 and ending at position 9. This annotation is considered true and has been annotated by multiple annotators and a computer. The associated ID is 'uniprot/3037912,' and it belongs to the category of "Genes & Molecular Sequences."

   Common Terms:

- Coreference: The data seems to involve annotations related to coreference resolution, where mentions that refer to the same entity are identified and linked together.
- Mentions: Refers to specific spans of text that are annotated.
- Clusters: Sets of coreferent mentions linked together.

Challenges:

- False Annotations: Some rows indicate false concepts or relations, and these might be instances where the annotators disagreed or made errors.
- Relation Annotations: There are rows indicating relations between annotated entities.

Entity Types:

- Genes & Molecular Sequences: Annotations related to genes, molecular sequences, and related concepts.
- Chemicals & Drugs: Annotations related to chemicals and drugs.

.

Tools needed to clean up biomedical datasets to extract genes, proteins, drugs, etc

Below are some of the tools that are used for extracting biomedical text entities such as genes and proteins in the research papers that we have gone through:

1. DNorm
2. ABNER
3. GNormPlus
4. BeCAS
5. BANNER
6. LINNAEUS
7. CRF
8. Stanford Named Entity Recognizer (NER)
9. MetaMap
10. NCBO Annotator
11. PubTator
12. BioBERT-Base v1.1
13. BioBERT-Large v1.1
14. ScispaCy
15. MetaMap [T21]
16. GeneView [T69]
17. SCAIView [T70]
18. Hidden Markov Models (HMMs)
19. Named Entity Normalization (NEN)
20. Automatically Extracted Dictionaries
21. Linnaeus
22. Toward Information Extraction

Selected Tools:

Linnaeus:
Linnaeus is designed to recognize and classify the scientific names of living organisms, including plants, animals, fungi, bacteria, and viruses, mentioned in textual documents. The tool is named after Carl Linnaeus, the renowned botanist who formalized the modern system of naming organisms, known as binomial nomenclature.

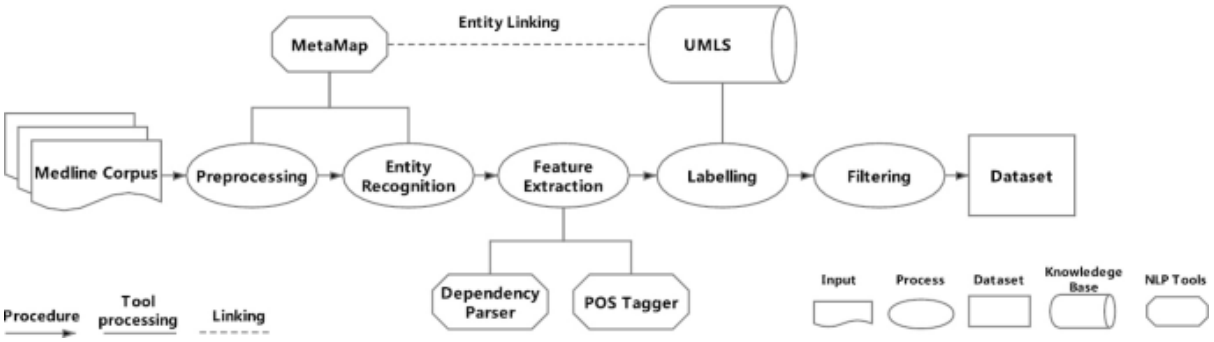
PubTator: (link)
PubTator provides annotations for PubMed articles, including gene and protein mentions. It's a valuable resource for extracting genes, diseases, and proteins from literature.

- PubTator is a tool that can be used to complement the entity spans of some datasets that do not include them.
- To use PubTator, the input text is first processed to identify relevant entities, such as gene and protein names.
- These entities are then mapped to their corresponding identifiers in the NCBI Entrez Gene and UniProtKB databases.
- Finally, the annotated text is returned in a standardised format that includes the entity type, entity identifier, and the position of the entity in the text.

MetaMap:
- MetaMap is used to map biomedical terms in text to concepts in the Unified Medical Language System (UMLS) Metathesaurus.
- This is done by analysing the syntax and semantics of the text and identifying relevant concepts from the UMLS Metathesaurus that match the text.
- The output of MetaMap is a list of concepts that are relevant to the text, along with their semantic types and other information. This information can be used to identify and extract biomedical entities such as genes, proteins, and diseases from the text.

Also it is important to note that the accuracy of these tools can be affected by factors such as the quality of the input text, the complexity of the concepts being identified, and the availability of relevant resources.

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-03889-5
BIOREL DATASET CREATION



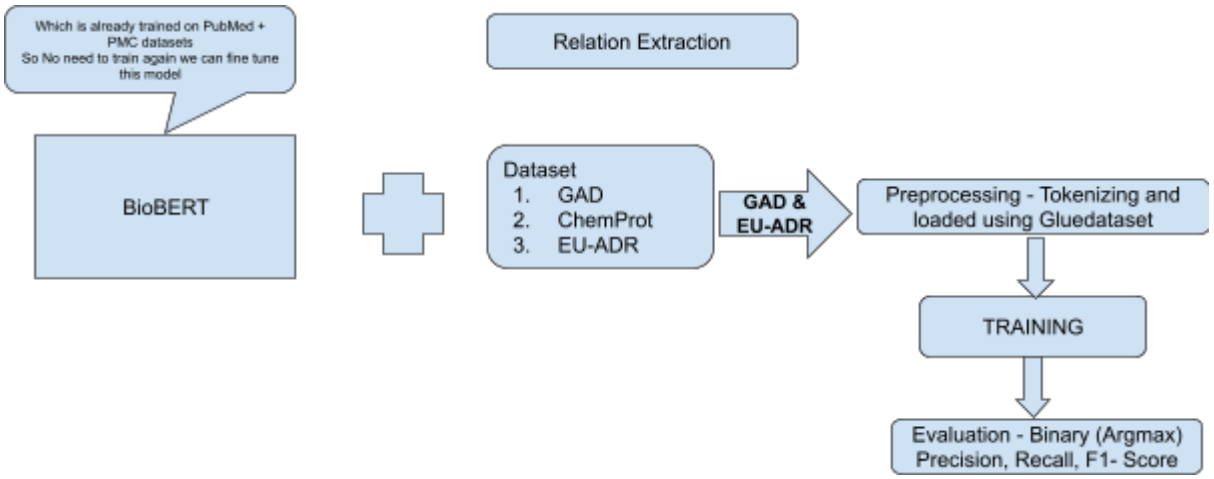
METHODOLOGY REVIEW

End-to-End Workflow

BioBERT
utilized the sentence classifier of the original version of BERT, which uses a [CLS] token for the classification of relations

BioBERT trained on PubMed and PMC datasets (trained on only words) 200K and 270K pre-training steps were optimal
Naver Smart Machine Learning (NSML) utilized for large-scale experiments that need to be run on several GPUs. We used eight NVIDIA V100 (32GB) GPUs for the pre-training.

In our case we need to fine tune BioBERT + RE classification task on top of it



Source -
https://github.com/dmis-lab/biobert-pytorch/blob/master/relation-extraction/run_re.py#L28

Evaluation

1. How do we evaluate the relation extraction classification models

GAD - Binary Classification

```
# binary
if args.task == "binary":
    pred = [preddf.iloc[i].tolist() for i in preddf.index]
    pred_class = [int(v[1]) for v in pred[1:]]

    p,r,f,s = sklearn.metrics.precision_recall_fscore_support(y_pred=pred_class, y_true=testdf["label"])
    results = dict()
    results["f1 score"] = f[1]
    results["recall"] = r[1]
    results["precision"] = p[1]
    results["specificity"] = r[0]
```

Chemprot (Classificatino of 5 classes)

```
# chemprot
# micro-average of 5 target classes
# see "Potent pairing: ensemble of long short-term memory networks and support vector machine for chemical-protein relation extraction
if args.task == "chemprot":
    pred = [preddf.iloc[i].tolist() for i in preddf.index]
    pred_class = [np.argmax(v) for v in pred]
    str_to_int_mapper = dict()

    for i,v in enumerate(sorted(testdf["label"].unique())):
        str_to_int_mapper[v] = i
    test_answer = [str_to_int_mapper[v] for v in testdf["label"]]

    p,r,f,s = sklearn.metrics.precision_recall_fscore_support(y_pred=pred_class, y_true=test_answer, labels=[0,1,2,3,4], average="micr
    results = dict()
    results["f1 score"] = f
    results["recall"] = r
    results["precision"] = p

for k,v in results.items():
    print("{:11s} : {:.2%}".format(k,v))
```

Strength of Relation

Output probabilities?

How to Create a Knowledge Graph

The nodes in a **knowledge graph represent entities, such as genes, proteins, diseases, and drugs**. The edges in a knowledge graph represent relations between entities, such as protein-protein interactions, gene-disease associations, and drug-drug interactions.

To build a knowledge graph on top of relation extraction, we can use the following steps:

1. Extract relations from biomedical text.
2. Cluster relations. Once we have extracted relations, we can cluster them together based on their similarity. This will help us to identify groups of related relations.
3. Identify entities. We can use natural language processing techniques to identify the entities that are involved in each relation.

4. Link entities to knowledge bases. We can link the entities in our knowledge graph to existing knowledge bases, such as PubMed and Wikidata. This will allow us to add additional information about the entities to our knowledge graph.
5. Visualize the knowledge graph. We can use visualization tools to visualize our knowledge graph and make it easier to explore and understand.

Once we have built a knowledge graph, we can use it to evaluate and understand the relations between entities. For example, we can use the knowledge graph to identify the most important relations in a particular domain, or to identify new and emerging relations. We can also use the knowledge graph to answer questions about the relations between entities, such as:

- What are the different types of relations between genes and diseases?
- What are the most common drug-drug interactions?
- What are the different ways that proteins interact with each other?

Knowledge graphs are a powerful tool for understanding the complex relationships between entities in the biomedical domain. By building a knowledge graph on top of relation extraction, we can make it easier to explore and understand the biomedical literature.

Here are some examples of how knowledge graphs can be used to evaluate and understand the relations between entities in the biomedical domain:

- A knowledge graph can be used to identify the most important genes involved in a particular disease. This information can be used to develop new diagnostic tests and treatments for the disease.
- A knowledge graph can be used to identify new and emerging drug-drug interactions. This information can be used to prevent patients from experiencing adverse side effects.
- A knowledge graph can be used to identify the different ways that proteins interact with each other. This information can be used to develop new drugs that target specific protein interactions.

Biomedical RE - University thesis report
https://oa.upm.es/71455/1/TFM_CHRISTIAN_MARLON_PANEQUE_MOREDA.pdf

PROJECT CODE REPO

Connect HPC

Terminal -1

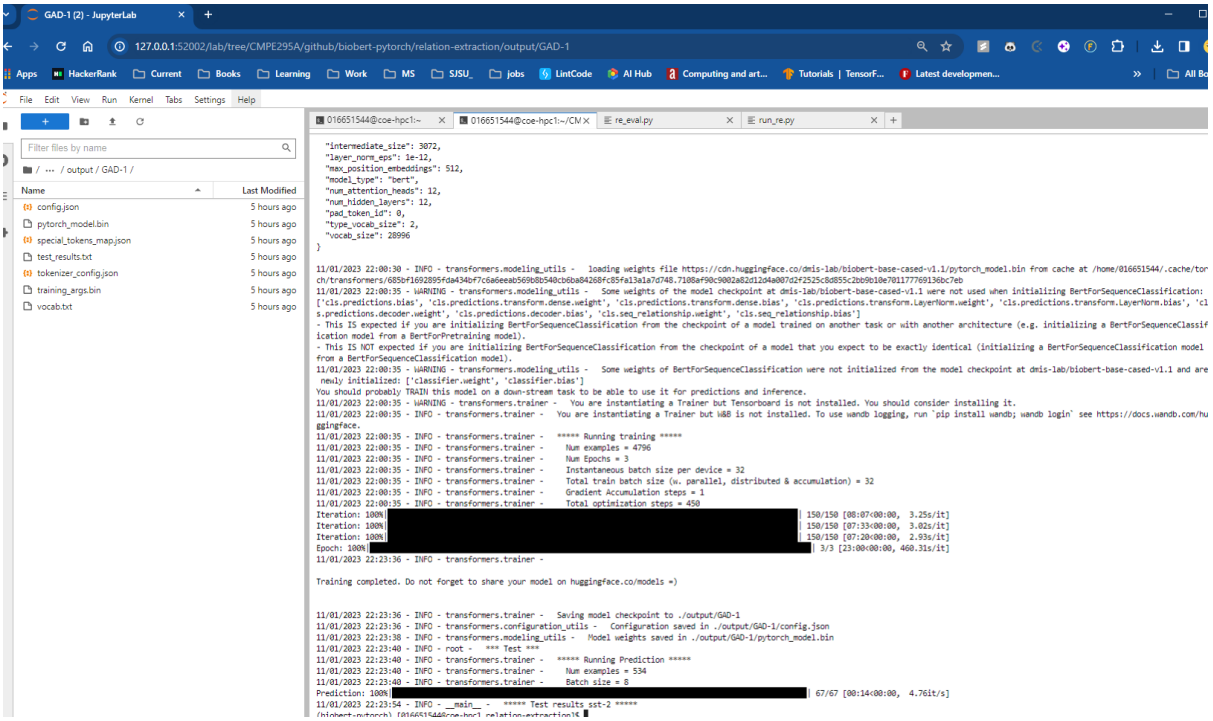
ssh 016651544@coe-hpc1.sjsu.edu
srun -p gpu -n 1 -N 1 -c 2 --pty /bin/bash
srun -n 1 -N 1 -c 4 --pty /bin/bash

Terminal -2
ssh -L 52001:localhost:52001 016651544@coe-hpc1.sjsu.edu
jupyter lab --no-browser --port=52001
Dataset loading - Dataset Visualize

BIOBERT PYTORCH (github - Link)

1. git clone https://github.com/dmis-lab/biobert-pytorch/
2. cd biobert-pytorch/
3. conda create -n biobert-pytorch python=3.7
4. conda activate biobert-pytorch
5. pip install transformers==3.0.0
6. Run ./download.sh
7. conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytorch -c nvidia
8. pip install chardet pandas
9. cd relation-extraction
    a. Follow these instructions - https://github.com/dmis-lab/biobert-pytorch/tree/master/relation-extraction
10. python run_re.py \
11. --task_name SST-2 \
12. --config_name bert-base-cased \
13. **--data_dir ${DATA_DIR} **
14. **--model_name_or_path dmis-lab/biobert-base-cased-v1.1 **
15. --max_seq_length ${MAX_LENGTH} \
16. --num_train_epochs ${NUM_EPOCHS} \
17. --per_device_train_batch_size ${BATCH_SIZE} \
18. --save_steps ${SAVE_STEPS} \
19. --seed ${SEED} \
20. --do_train \
21. --do_predict \
22. --learning_rate 5e-5 \
23. --output_dir ${SAVE_DIR}/${ENTITY} \
24. --overwrite_output_dir
25. Total time for training - 22:34 minutes
26.

Training




```
(biobert-pytorch) [016651544@coe-hpc1 relation-extraction]$ python ./scripts/re_eval.py --output_path=${SAVE_DIR}/${ENTITY}/test_results.txt -
-answer_path=${DATA_DIR}/test_original.tsv
f1 score    : 81.97%
recall      : 88.97%
precision   : 75.99%
specificity : 68.77%
```

BIOBERT - EUADR training (main repo)

Steps

1. git clone https://github.com/dmis-lab/biobert
2. cd biobert; pip install -r requirements.txt
3. conda create -n biobert
4. conda activate biobert
5. conda install python=3.7
6. pip install -r requirements.txt
7. pip install pandas
8. pip install tensorflow==1.15
9. conda install numpy~=1.19.5
10. wget https://github.com/naver/biobert-pretrained/releases/download/v1.0-pubmed-pmc/biobert_v1.0_pubmed_pmc.tar.gz

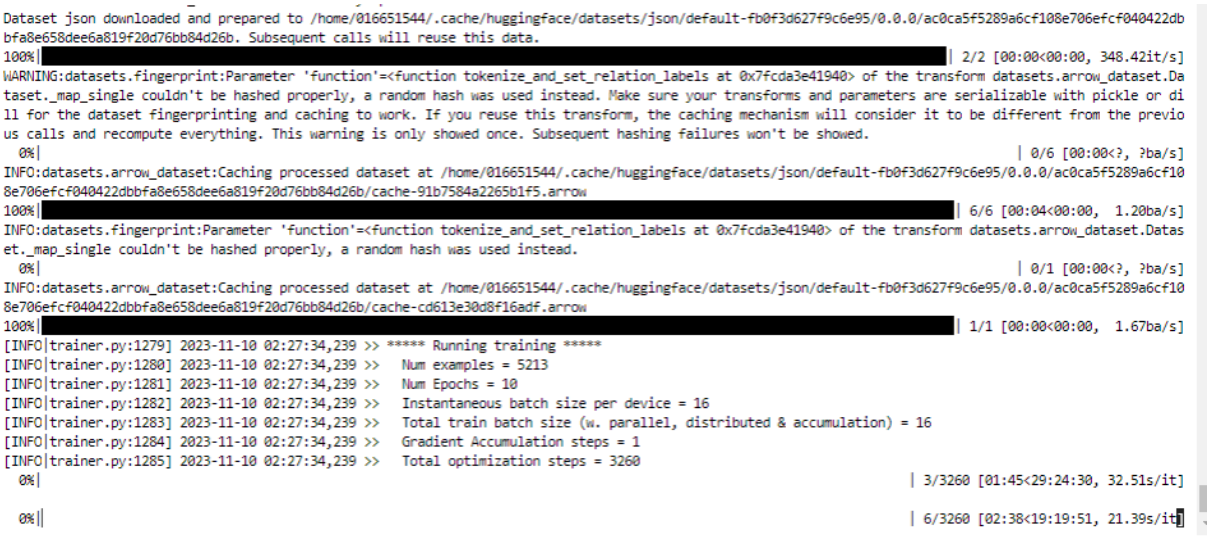
11. tar -xf biobert_v1.0_pubmed_pmc.tar.gz
12. export BIOBERT_DIR=./biobert_v1.1_pubmed_pmc
13. echo $BIOBERT_DIR
14. export TASK_NAME=euadr
15. export RE_DIR=./datasets/RE/euadr/1
16. export OUTPUT_DIR=./re_ouptut_euadr
17. python run_re.py --task_name=$TASK_NAME --do_train=true --do_eval=true --do_predict=true --vocab_file=$BIOBERT_DIR/vocab.txt --bert_config_file=$BIOBERT_DIR/bert_config.json --init_checkpoint=$BIOBERT_DIR/biobert_model.ckpt.data-00000-of-00001 --max_seq_length=128 --train_batch_size=32 --learning_rate=2e-5 --num_train_epochs=3.0 --do_lower_case=false --data_dir=$RE_DIR --output_dir=$OUTPUT_DIR > euadr_output.txt 2>&1 &

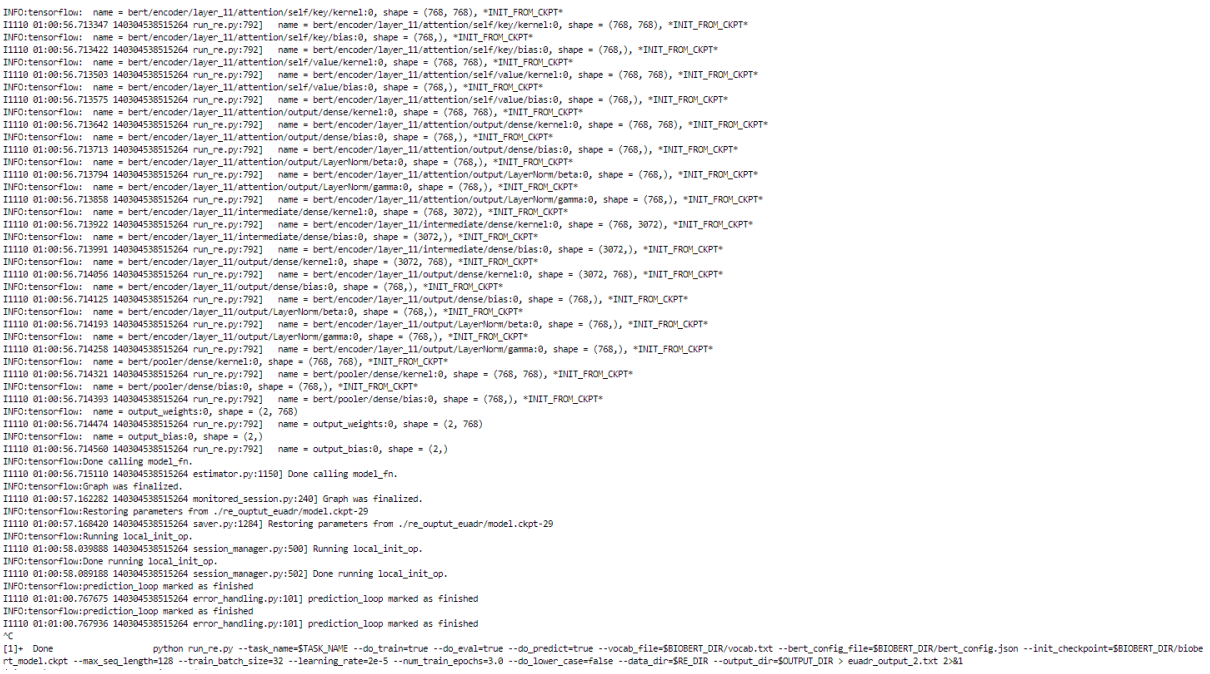
```
I1109 23:56:47.403966 139671310591808 basic_session_run_hooks.py:541] Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
I1109 23:56:50.250227 139671310591808 monitored_session.py:240] Graph was finalized.
2023-11-09 23:56:50.250633: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2023-11-09 23:56:50.265192: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 1999985000 Hz
2023-11-09 23:56:50.266212: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x88b9dd0 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2023-11-09 23:56:50.266286: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
2023-11-09 23:56:50.306817: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcuda.so.1
2023-11-09 23:56:50.420598: E tensorflow/stream_executor/cuda/cuda_driver.cc:318] failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
2023-11-09 23:56:50.420685: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (coe-hpc1.sjsuad.sjsu.edu): /proc/driver/nvidia/version does not exist
INFO:tensorflow:Running local_init_op.
I1109 23:57:07.964576 139671310591808 session_manager.py:500] Running local_init_op.
INFO:tensorflow:Done running local_init_op.
I1109 23:57:08.196357 139671310591808 session_manager.py:502] Done running local_init_op.
INFO:tensorflow:Saving checkpoints for 0 into ./re_ouptut_euadr/model.ckpt.
I1109 23:57:14.435770 139671310591808 basic_session_run_hooks.py:606] Saving checkpoints for 0 into ./re_ouptut_euadr/model.ckpt.
INFO:tensorflow:global_step/sec: 0.0523624
I1109 23:58:09.366435 139671310591808 tpu_estimator.py:2307] global_step/sec: 0.0523624
INFO:tensorflow:examples/sec: 1.6756
I1109 23:58:09.437762 139671310591808 tpu_estimator.py:2308] examples/sec: 1.6756
INFO:tensorflow:global_step/sec: 0.0373234
I1109 23:58:36.150560 139671310591808 tpu_estimator.py:2307] global_step/sec: 0.0373234
INFO:tensorflow:examples/sec: 1.19435
I1109 23:58:36.154683 139671310591808 tpu_estimator.py:2308] examples/sec: 1.19435
INFO:tensorflow:global_step/sec: 0.109584
I1109 23:58:45.269227 139671310591808 tpu_estimator.py:2307] global_step/sec: 0.109584
INFO:tensorflow:examples/sec: 3.50669
I1109 23:58:45.270830 139671310591808 tpu_estimator.py:2308] examples/sec: 3.50669
INFO:tensorflow:global_step/sec: 0.0384973
I1109 23:59:11.251044 139671310591808 tpu_estimator.py:2307] global_step/sec: 0.0384973
INFO:tensorflow:examples/sec: 1.23191
I1109 23:59:11.253491 139671310591808 tpu_estimator.py:2308] examples/sec: 1.23191
INFO:tensorflow:global_step/sec: 0.1159
I1109 23:59:19.872989 139671310591808 tpu_estimator.py:2307] global_step/sec: 0.1159
INFO:tensorflow:examples/sec: 3.70881
I1109 23:59:19.874024 139671310591808 tpu_estimator.py:2308] examples/sec: 3.70881


INFO:tensorflow:global_step/sec: 0.0463687
I1109 23:59:41.440524 139671310591808 tpu_estimator.py:2307] global_step/sec: 0.0463687
INFO:tensorflow:examples/sec: 1.4838
I1109 23:59:41.441865 139671310591808 tpu_estimator.py:2308] examples/sec: 1.4838
INFO:tensorflow:global_step/sec: 0.103717
I1109 23:59:51.081251 139671310591808 tpu_estimator.py:2307] global_step/sec: 0.103717
INFO:tensorflow:examples/sec: 3.31894
I1109 23:59:51.082630 139671310591808 tpu_estimator.py:2308] examples/sec: 3.31894
```

PPI - Relation Extraction (link)

- conda create -n PPI-RE
- conda activate PPI-RE
- conda install python=3.9
- pip install -r requirements.txt
- export DATASET_DIR=./datasets
- export OUTPUT_DIR=./output
- export DATASET_NAME=PPI/original/AImed
- export SEED=1
- python src/relation_extraction/run_re.py --model_list dmis-lab/biobert-base-cased-v1.1 --task_name "re" --dataset_dir $DATASET_DIR

--dataset_name $DATASET_NAME --output_dir $OUTPUT_DIR --do_train --do_predict --seed $SEED --remove_unused_columns False --save_steps 100000 --per_device_train_batch_size 16 --per_device_eval_batch_size 32 --num_train_epochs 10 --optim "adamw_torch" --learning_rate 5e-05 --warmup_ratio 0.0 --weight_decay 0.0 --relation_representation "EM_entity_start" --use_context "attn_based" --overwrite_cache --overwrite_output_dir





Code references

1. **Datasets -**https://huggingface.co/bigbio
2. **Code repo:-** https://github.com/BNLNLP/PPI-Relation-Extraction
3. **Benchmarking-** https://github.com/zjunlp/LREBench

4. **Notebook** to run a model on biomedical text - https://github.com/NVIDIA/NeMo/blob/main/tutorials/nlp/Relation_Extraction-BioMegatron.ipynb
5. https://github.com/dmis-lab/biobert
6. https://maayanlab.cloud/Harmonizome/dataset/GAD+Gene-Disease+Associations
7. Can be used for Baseline testing - **https://github.com/dmis-lab/biobert-pytorch/tree/master/relation-extraction**

NEXT STEPS

10/20/2023 - Next Steps

2. How do we evaluate the relation extraction classification models
3. How the models work -> end-end workflow overview
4. How can we extract the strength of the relation between entities
5. Interested in Writing Paper ? - Yes

11/02/2023 - Next Steps

1. Looking at architecture (BioBERT) - Understand.
2. Multi relations within a sentence as input to model (check if any existing papers are available) - Think about if it makes sense/ helpful
3. BioRED/Chemprot - Run on BioBERT.\

Updates -

Trained BioBERT on EU-ADR dataset and PPI (on HPC)

BioBERT Architecture in-depth (https://github.com/dmis-lab/biobert/blob/master/modeling.py)

BioREx - paper (improving RE by Heterogenous datasets)

11/10/2023 - Next Steps

1. Understand method for BioREx
2. How can we process document level biomedical text (multi relation)
3. Understand datasets - , AIMed [1], DrugProt [2], DDI [9], HPRD50 [10], and BC5CDR [13])
4. Motivation behind the project - which scenarios is this being used?

BioBERT Architecture:

BERT (Bidirectional Encoder Representations from Transformers), a transformer-based model for natural language processing tasks, introduced by Google. The architecture is implemented using TensorFlow:

1. **BertConfig**: A class to store configuration parameters for the BERT model, such as vocabulary size, hidden layer size, number of hidden layers, number of attention heads, etc. This class allows for easy customization of the BERT model.

2. **BertModel**: The core class that implements the BERT model. It encapsulates the entire model architecture and functionality. Key components include:

- **Embeddings**: The input text is first converted into embeddings. This includes word embeddings, positional embeddings, and segment embeddings (for distinguishing different sentences).
- **Encoder**: The core of BERT, consisting of multiple layers of transformer blocks. Each block includes multi-head self-attention and a feed-forward neural network. This part is responsible for understanding the context and relationships between words in a sentence.
- **Pooler**: Extracts a fixed-size feature vector from the transformer's output, typically by taking the output corresponding to the first token (often [CLS] token) for classification tasks.

3. **Helper Functions:** The code includes several utility functions:

- Activation functions like GELU (Gaussian Error Linear Unit).
- Functions to create attention masks, perform dropout, layer normalization, etc.
- Functions to reshape tensors and verify tensor shapes.

4. **Transformer Model Implementation**: The core transformer model is implemented in the transformer_model function. It sequentially applies each transformer layer (attention followed by normalization and feedforward network) to the input.

5. **Attention Mechanism**: The attention_layer function implements the multi-head self-attention mechanism, which allows the model to weigh the importance of different words in a sentence.

6. **Embedding Layers**: The embedding_lookup and embedding_postprocessor functions are responsible for converting input tokens into embeddings, adding positional and segment embeddings, and performing necessary post-processing like dropout and layer normalization.

The above describes the Biobert model. However, in this specific github repo, the model is finetuned on GAD dataset for RE.