Coreference resolution for biomedical text mining using domain-specific pretrained transformer models

Smit Jayesh Mirani
(2571130M)

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

13th December 2021

Abstract

This project focuses on the coreference resolution task i.e., linking an antecedent with its mention in an article over longer spans of sentences. In this project, we particularly focus on the Biomedical text present in the CRAFT Shared Tasks 2019. The CRAFT corpus is challenging as the reference chains can be discontinuous and may also span across the entire length of the article. We plan to examine domain-specific language models, by leveraging deep learning models (PubMedBERT, BioBERT, SciBERT, ClinicalBERT) that have been pre-trained on separate domain-specific text for our coreference resolution task. Our aim is to evaluate the performances of domain-specific pre-trained model on a coreference task compared to a baseline model (BERT Base) that has been trained on generic textual content, using F1 score metric.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

Name: Smit Jayesh Mirani (student) Signature: Smit Mirani

Acknowledgements

I would like to thank Dr. Jake Lever for his constant support, encouragement, and guidance throughout the dissertation. His motivation and direction in choosing the project, understanding the domain of NLP and providing relevant resources from time to time was invaluable. I would also like to thank my family for their constant moral support.

Contents

Chapter 1 Introduction

1.1 Motivation

Natural Language Processing (NLP) and its applications are increasing every day. With big data growing even bigger, increasing contributions from text messaging apps, social media, forum, news, published papers, blogs, etc. producing enormous amount of text data every second, it is very important to make sense of everything [1]. With the kind of unstructured data that is generated in the written and spoken language, NLP helps resolving these ambiguities by adding useful numeric structures.

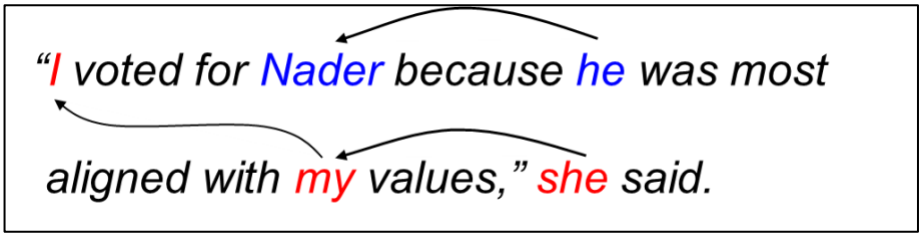

Figure 1: Example of coreference resolution in general text provided by The Stanford Natural Language Processing Group [16]

Coreference resolution sits at the core of NLP, by determining the linguistic relations between entities and their mentions in natural language. It is the most challenging task and supports advanced NLP tasks such as question answering, text summarization etc. Figure 1 describes the concept of coreference resolution, where "I" and "Nader" are the entities and "he", "my" and "she" are the mentions of the entitites [16].

Rod and cone photoreceptors subserve vision under dim and bright light conditions, respectively. The differences in **their** function are thought to stem from **their** different gene expression patterns, morphologies, and synaptic connectivities.

Figure 2: Example of coreference relations in the CRAFT-CR 2019 dataset [17]. Text marked in red is the entity and ones in green are its references

Particularly in the field of Biomedical domain, coreference resolution has become one of the most challenging and important tasks to extract

information from the complex articles [3]. Figure 2 shows that *"Rod and cone photoreceptors"* are the entities being talked about and it is referred by co-references *"their"*. Someone would need some background information to understand that different photoreceptors have different gene expression patterns. While it is easy for humans to understand the semantics and relations of coreferences, it is not so easy for machines. This is where coreference resolution using Natural Language Processing comes into picture.

Many coreference resolution models have been proposed. While there has been much research done in this field, various kinds of rule based, machine learning and even deep learning models have been proposed, we focus on analyzing and reviewing the effect of using deep learning models pre-trained on different domains for single corpus of biomedical texts and their embeddings. We compare the results obtained from our study against the gold standard results provided by the task to see if these transfer learning methods do provide a substantial advantage.

1.2 Purpose

Although much research has been done in the field of coreference resolution in general domain, using rule and heuristic based approaches, in recent years much of the focus has been shifted to using Deep Learning based Language representation models [3]. Further diving deep, new language representation models have been developed by pre-training them on domain specific corpora. To put into context, models such as BioBERT [6], Bio_ClinicalBERT [7], SciBERT [8],PubMedBERT etc. have been developed by pretraining them on a domain specific corpora and each research shows the improvement in the performance of their respective NLP tasks over different corpora. However, in this study we aim to select a single corpora and use each of these above models to study the actual performance gain in coreference resolution task against the gold standard provided available. The purpose of this study is to check the cross-domain performances of thes ePre-trained language models (PLM) on a single corpora of CRAFT-CR shared file.

1.3 Summary

The structure of this report is as follows. In chapter 2 we briefly provide a background and critical evaluation of the work that has been done previously and the advancements that have been made in the field of coreference resolution in general and using transfer learning. Alongside this we also briefly discuss the transformer models and their architecture that are employed to obtain desired results. In chapter 3, we showcase our

implementation strategies, which includes a description of the CRAFT-CR task 2019 and with a structured examination of the corpus that we have used. This chapter also focuses on the challenging data pre-processing task, the algorithm used in our evaluation of this study and shed some light on the challenges faced during this analysis. Moving on in chapter 4 we discuss the evaluation of our results using key metrics & different data visualization techniques. Further, in chapter 5, we summarize our findings, limitations of our study and conclude with some suggested future work.

Chapter 2 Survey

As discussed in previous chapter, much work has been done already in the field of NLP and to be specific, coreference resolution which sits in the heart of NLP to aide some complex tasks like summarization, Question Answering, text classification etc. In this chapter we will focus more on previous works and different methods used for coreference resolution in the Biomedical domain and subsequently how the field has evolved in the last decade.

2.1 Historical Studies

P.Lu et al. in their survey have discussed different biomedical language representation models, and different coreference models in this domain [3]. They explain the evolution of these models from rule-based models, which used very confined semantic & syntactic rules to achieve coreference resolution, which were limited to certain corpus and relations only. They have also discussed how after the public availability of BioNLP 2011 protein coreference dataset [11], the field saw an uprising of many machine learning approaches, which were essentially trained to classify the anaphora and antecedent. However, with the advancements, a hybrid of rule based and machine learning approaches was developed that proved to improve the performance on the task [12].

In the recent years, with surge of research in Deep Learning methods, the field of coreference resolution has not been left untouched. Clark and Manning in their research [13] produced impressive models that incorporated neural network approach which produced high dimensional vector representations of clusters of mentions. But they required overhead resources to achieve good performance. In 2017, Lee et al. came up with a very first of its kind end-to-end deep learning coreference resolution model that used LSTM (Long Short Term Memory model) [5].

2.2 Advancements in Coreference Resolution

The method of transfer learning i.e. using knowledge of pre trained machine learning model for different tasks, has proven to be very successful. Its applications are enormous and are being viewed as the future of machine learning. Transfer learning takes very less data and computational resources [15].

Coreference resolution in recent years has also seen the applications of transfer learning. Particularly for extracting contextual embeddings from text, the Bidirectional Encoder Representations from Transformers [BERT] model [9] has become a prime entity and has laid foundation for domain specific NLP tasks. Ever since Google Launched BERT [9] in 2018, there

have been numerous adaptations of the model for various NLP tasks such as hate speech detection, sentiment analysis, search engine optimization etc. In 2019, Trieu et al. went a step further to address the issue of coreference resolution in full text article corpus such as the CRAFT-CR dataset [14]. Their key contributions were filtering out noisy mention spans and using BERT instead of LSTM model. The performance improved more on full span text documents like the CRAFT Dataset.

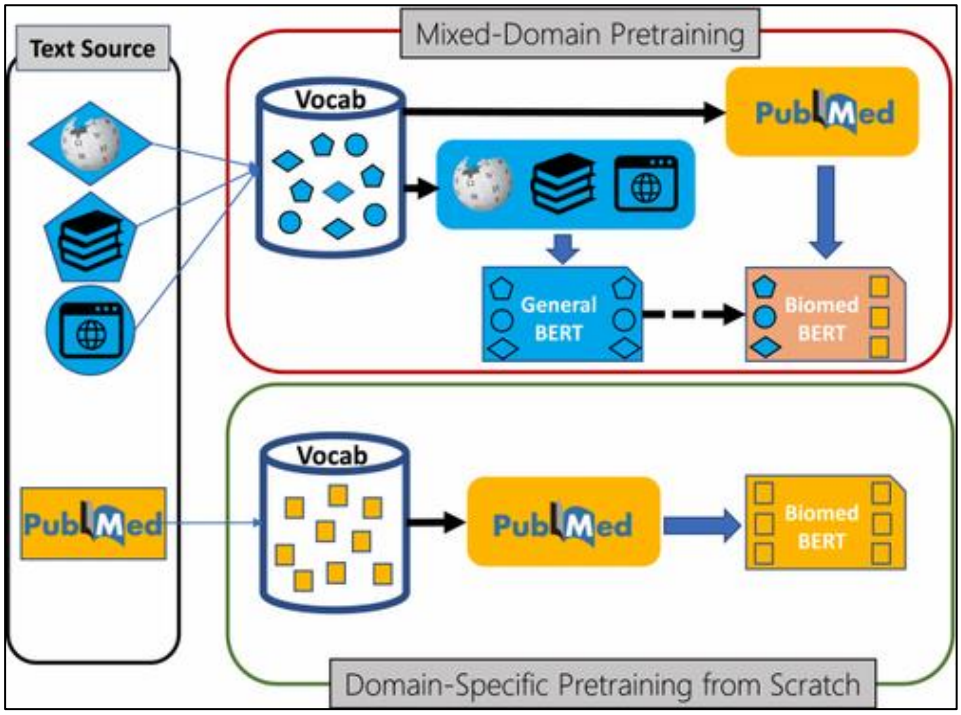

Figure 3 : Two representations of Transfer Learning mechanisms as described by Y. Gu et al. [10]

However, there are different approaches to Transfer Learning as well as explained by Y. Gu et al at Microsoft Research [10]. As we can see in Figure 3, top, first is a mixed-domain pretraining method i.e. continual learning method. Here we continue domain specific training on a model that has previously been trained on general corpora like BERT, which has been trained on Wikipedia and BookCorpus. The second method, Figure 3, bottom, is an approach where a model is pre-trained from scratch with domain specific corpora rather than continuing pre-training from general corpora. They prove that the second method outperforms the first method for wide range of biomedical NLP applications, including but not limited to Coreference Resolution. This approach has been applied by various models as discussed in section 2.1, where several models have been proposed and have been trained from scratch on different corpora depending on the task at hand and they have shown promising improvements, specifically in the coreference resolution task.

2.3 Reflections

After detailed literature survey, analysis and studying the history of coreference resolution, in general and in biomedical domain, learning the evolution of coreference resolution from syntactic & semantic rule based methods to machine learning classification methods to the model deep learning methods that include Neural Network approaches, applications of state of the art end-to-end Transformer Models to the most advanced applications of the BERT model and adaptations, this field has seen great advancements. This can be seen in the applications of Coreference Resolution in Natural Language Processing like text summarization, Question Answering, Chat Bots, Search engine optimization, text generation etc. In this study, what we focus on is the behaviour of all these Pre-trained deep learning models on a single corpora. We need to evaluate the behaviour of all these models on one of the domain specific corpora to see if after being pre-trained, how do they behave and is all their behaviour same for all.

Chapter 3 Implementation

In this chapter we go through a detailed description of the dataset of the CRAFT-CR 2019 shared task [19] corpus, the data collection, data preprocessing, model selection, pseudo-code to evaluate different model's performances over the same corpus and challenges faced in this process.

3.1 The Dataset

In this study, we use the Colorado Richly Annotated Full Text (CRAFT) Corpus v3.1.3 [17]. The corpus has 67 full text biomedical articles about mouse genomics, which have entities with referents/mentions in the same document [18]. Each article is a part of the huge PubMedCentral Open access subset, which itself includes 3.4 million journal articles. Some stats about the CRAFT data set are :-

	Statistics
No. of Documents	67
No. of sentences	21,731
Average sentences per doc	324.34
No. of mentions	77,755
No. of discontinuous mentions	4485
No. of coreferences	16,302

Table 1. Characteristics of the CRAFT Corpus [14]

```
BRCA2 was the second breast cancer susceptibility gene to be discovered, and was
isolated through positional cloning using data from families with inherited
breast cancer [4]. Cells with mutant BRCA2 protein are, like many cancer cells,
genetically unstable and accumulate gross chromosomal rearrangements [5,6]. The
sequence of this large protein (3418 amino acids) offers very little clue to its
function, although there are eight repeated segments (termed BRC repeats) in the
middle of the protein that are highly conserved among mammalian orthologs [7,8].
```

Figure 4. A paragraph extracted from document PMC11597317 of CRAFT corpus [17]

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<knowtator-project>
    <document id="11597317" text-file="11597317.txt">
        <annotation annotator="Annotator" id="0" type="identity">
            <class id="IDENTITY chain" label="IDENTITY chain"/>
            <span end="5" id="11597317-0" start="0">BRCA2</span>
        </annotation>
        <annotation annotator="CCP Colorado Computational Pharmacology, UC Denver" id="11597316-KC_Instance_1" type="identity">
            <class id="Noun Phrase" label="Noun Phrase"/>
            <span end="5" id="11597317-1" start="0">BRCA2</span>
        </annotation>
        <annotation annotator="Annotator" id="1" type="identity">
            <class id="IDENTITY chain" label="IDENTITY chain"/>
            <span end="34" id="11597317-2" start="10">homologous recombination</span>
        </annotation>
        <annotation annotator="CCP Colorado Computational Pharmacology, UC Denver" id="11597316-KC_Instance_10085" type="identity">
            <class id="Noun Phrase" label="Noun Phrase"/>
            <span end="34" id="11597317-3" start="10">homologous recombination</span>
        </annotation>
        <annotation annotator="Annotator" id="2" type="identity">
            <class id="IDENTITY chain" label="IDENTITY chain"/>
            <span end="63" id="11597317-4" start="46">Two recent papers</span>
        </annotation>
        <annotation annotator="CCP Colorado Computational Pharmacology, UC Denver" id="11597316-KC_Instance_20006" type="identity">
            <class id="Noun Phrase" label="Noun Phrase"/>
            <span end="63" id="11597317-5" start="46">Two recent papers</span>
        </annotation>
        <annotation annotator="CCP Colorado Computational Pharmacology, UC Denver" id="11597316-KC_Instance_3" type="identity">
            <class id="Noun Phrase" label="Noun Phrase"/>
            <span end="152" id="11597317-6" start="109">the breast cancer susceptibility gene BRCA2</span>
        </annotation>
        <annotation annotator="Annotator" id="3" type="identity">
            <class id="IDENTITY chain" label="IDENTITY chain"/>
            <span end="146" id="11597317-7" start="113">breast cancer susceptibility gene</span>
        </annotation>
```

Figure 5. The coreference annotations snippet from the knowtator files present in the corpus for document PMC11597317 [17]

The corpus provides us with full text articles and also their corresponding annotations in the knowtator format using the OntoNotes annotation scheme. Here we can see in figure 5, a textual snippet of one of the 67 full text documents. In figure 5, we see a snippet of a file, which represents the coreferences in a knowtator format which is a general-purpose text annotation tool [18]. This format provides a mapping between the entity and its mentions. This format contains many vital information such as the mention, its span and other meta data about the annotation.

3.2 Data Preprocessing

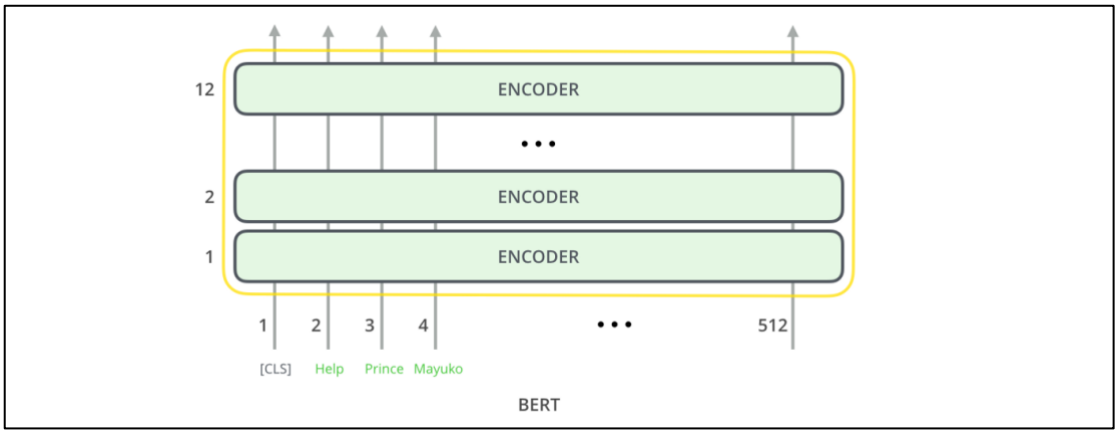

Figure 6. A high level view of a general BERT Model [22]

Since the introduction of the BERT model, the bidirectional transformer based language model, which uses encoders instead of decoders, it has revolutionized the NLP domain. Figure 6 shows an eagle's eye view of the

BERT Model. The BERT Base model has 768 hidden layer size, which takes in a token/word or sentence or even larger chunk of text to provide a contextual token embedding per word. These token embeddings are vocabulary IDs for each token/word in the context of that particular sentence.

In order to extract the contextual word embeddings of each word in the original text, we decided to process one sentence at a time as input to the BERT model. However, the only raw data we have is either the full text article of the Knowtator format that has the annotation information. This particular task expects us to have our input files in a CoNNL format [19].

To achieve this task, we use CRAFT group's Docker image [20,21] that converts the knowtator format represented in the Figure 5 to a CoNLL format as represented in Figure 7 below :-

```
509305  0   1   Spontaneous JJ  -   -   -   -   -   -   -   (1b|(2|(3a
509305  0   2   Autoimmunity    NN  -   -   -   -   -   -   -   -
509305  0   3   in  IN  -   -   -   -   -   -   -   3a)
509305  0   4   129 CD  -   -   -   -   -   -   -   (5|(4)|(6d)|1b)
509305  0   5   and CC  -   -   -   -   -   -   -   -
509305  0   6   C57BL   NN  -   -   -   -   -   -   -   (3a|(7|(8
509305  0   7   /   HYPH    -   -   -   -   -   -   -   -
509305  0   8   6   CD  -   -   -   -   -   -   -   7)
509305  0   9   Mice    NNS -   -   -   -   -   -   -   (1b)|(6d)|8)|5)|3a)|2)
509305  0   10  -   :   -   -   -   -   -   -   -
509305  0   11  Implications    NNS -   -   -   -   -   -   -   -
509305  0   12  for IN  -   -   -   -   -   -   -   -
509305  0   13  Autoimmunity    NN  -   -   -   -   -   -   -   (9)
509305  0   14  Described   VBN -   -   -   -   -   -   -   -
509305  0   15  in  IN  -   -   -   -   -   -   -   -
509305  0   16  Gene    NN  -   -   -   -   -   -   -   (10
509305  0   17  -   HYPH    -   -   -   -   -   -   -   -
509305  0   18  Targeted    VBN -   -   -   -   -   -   -   -
509305  0   19  Mice    NNS -   -   -   -   -   -   -   10)
```

Figure 7. A CoNLL file representation obtained after running the CRAFT group's Docker Image [20,21] on google cloud shell

We pass the knowtator files for all 67 files available in the corpus, and obtain the 67 CoNLL files for each document.

In this format, we have one word per row of the file and some metadata per column for each word. For our task, we are concerned with the column number 3 and column number 12 (last column), where column 3 is the labeled as 'Lemma' or the stem form of the text, which is basically an individual word or character in the file and column 12 stores the coreference information that stores the identity chain details.

Identity chain links are represented by a alpha-numeric values encapsulated in brackets. Set of words that come under same alphanumeric character come represent same entity. There are different types of identity

chain links (mentions). We can understand them by taking the above Figure 7 as an example :-

1. **Contiguous Mentions** :- A set of tokens that are encapsulated under same identity chain sequentially. In Figure 7, the examples of Contiguous mentions are :

    - **"Spontaneous Autoimmunity in 129 and C57BL / 6 Mice"** is a member of identity chain #2 which starts at token 1 & ends at token 9 (inclusive)
    - **"Gene – Targeted Mice"** is a member of identity chain #10 which starts at token 16 & ends at token 19 (inclusive)

2. **Singleton Mention** :- A single word representing a separate entity. In Figure 7, an example of the Singleton Mention is :-

    - **"Autoimmunity"** is a member of identity chain #9.

3. **Discontiguous Mentions** :- A set of tokens that are encapsulated under same identity chain, but they need not be sequential and can span across the document length. These are denoted by additional character after the integer identifier [19]. The task suggests there is no limit to the number of characters that can be appended to the integer value unless the integer comes first. In Figure 7, an example of discontiguous mentions is :-

    - ""**Spontaneous Autoimmunity in … C57BL / 6 Mice"** are members of identity chain #3 which includes tokens [1-3] and [6-9] inclusive.
    - **"129 … Mice"** tokens are members of identity chain #6 which includes token numbers 4 and 9.

Now, that we have our data preprocessing is done, we are ready to to use these CoNLL files as inputs to our BERT model and its various adaptations as discussed earlier in section 1.2.

3.3 Implementation Details

In this section we will discuss the algorithm, tools and pseudocode that was required for the study. Here we describe how we use the Pre-processed files to obtain the tokens and their embeddings for each of the five models selected Vis-à-vis **SciBERT, BioBERT, ClinicalBERT, PubMedBERT, BERT Base**.

We then move on to grouping the words that belong to one of the 3 identity chains mentioned above. We then predict the words that are very much similar to each other using cosine similarity between each word's contextual embedding vectors and compare that with the gold standard provided by

the task and calculate F1 scores per model for various threshold values of the cosine similarity.

3.3.1 Tools and Infrastructure

To successfully conduct our experiment we leveraged various Python libraries that are included in the Data Science stack and deep learning specially for NLP process such as :-

- Python 3.7
- Pytorch 1.10
- Hugging Face Transformer module v4.13.0
- Pygraphviz 1.7
- Pandas 1.1.5
- NumPy 1.19.5
- Matplotlib 3.2.2
- ScikitLearn 1.0.1

To keep the code and files updated without losing the progress, we use the Google Colab notebook along with google

3.3.2 Data Extraction, Transformation & Loading

Here we noticed that the input CoNLL file has many whitespaces and blank lines. Hence we clean out the files and remove any and all whitespaces to ready the file for loading. Looking at the structure of pre-processed data in CoNLL format, we decided to store the file in a Pandas DataFrame.

Tokenization and Token Embeddings extraction

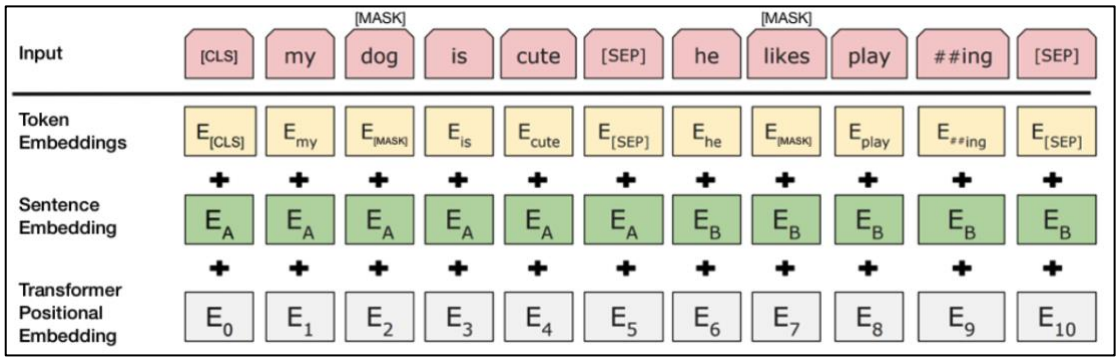

Figure 8. Sample input to the BERT Model [9]

Figure 8 explains the input format that BERT or any of its adaptations expect. In our study, all the models expect their input to have atleast 3 things:-

- Tokenized version of every word in the sequence. The number and types of tokens generated per word can be different from model to model depending on how it has been pretrained.

- **[CLS] token**: The classification token, must be the first token of sequence.
- **[SEP] token**: This is a delimiter token, which identifies the separations between different sequences.

3.3.3 The Algorithm

1. Below, Figure 9, is the snippet of the algorithm used. We have split the 67 files into :-
   - Training set of 40 files
   - Testing set of 27 files

2. Here, since we are using Google Colab we load all the pre-processed CoNLL files in Google Drive at a path "MyDrive/Sem3 Project/CoNLL files/", after giving necessary permissions to Colab for accessing the Drive so that a common ecosystem on cloud is maintained consistently.

3. The algorithm below shows in detail how for every model we extract tokens and contextual word embeddings for each sentence in the CoNLL files of training set.
   Note: We got help from an article on Towards Data Science portal to help us extract the embeddings and tokens from a BERT model, the reference for which has been provided in code as well [15]

4. We then combine the tokens and their embeddings per mention into groups, find the mean context vector per group.

5. We move on to calculate the "all pairs cosine similarity adjacency matrix", with cosine similarity threshold values ranging from 0.85 to 1.00 with increment of 0.01.

6. We also calculate a "Gold adjacency matrix" where we create an edge (cell value=1) between the groups that have same mention numbers.

7. The aim is to calculate True Positives (TP), False Positives (FP), True Negatives (TN) & False Negatives (FN) for all 5 models, for every file and for every increment in the threshold values from the confusion matrix.

8. Finally, we calculate the F1 scores for every model at various threshold values from the TPs, TNs, FPs and FNs obtained above.

9. We then select the cosine similarity threshold value to create the "all pairs cosine similarity adjacency matrix" that gives the highest F1 score and then use that fixed threshold to carry on all the steps above except variable threshold (except step 4) for remaining 26 files in test domain.

The snippet of the algorithm below is a detailed explanation of all the above steps. After getting the F1 scores, we select a sample from a file to generate the un-directed graphs of both predicted and gold standard mention links. We do this because, the original files are of thousands of lines and the Networkx tool consumes lots of resources and time to create a connected graph for files these long. Therefore, we believe a sample should be sufficient for visualization purpose.

Note: To generate the connected graphs, we must only pass one file that contains a small passage from one of the CoNLL files in test domain, with single threshold for cosine similarity obtained from the training phase, through the main code file.

```
for each model in ("sciBert","bioBert","clinicalBert","PubMedBERT_fulltext","BERT_base_uncased"):
  for each file in the directory of CoNLL files (67 files):
    create a new dataframe with columns in ["words", "coref","tokens_per_word","embedding_per_token"]
    store the column no 3 and 12 from read file in new dataframe's columns 1 and 2

    read the words one by one from column ["words"] of dataframe and
    store it as a sentence string in texts list until we encounter a delimiter fullstop.

    for every sentence in texts list:
      prepare the sentence for input to BERT model
      pass the sentence as an input to the tokenizer.
      Get tokenized text (per word) and tokens tensor as output

      pass this tokens tensor to get bert embeddings per token
      Get a contextual embeddings vector of length 768 per token (per word of sentence)

      Append the tokenized text and contextual embeddings vector in original dataframe

    for words with more than 1 tokens per word:
      find the mean vector of all the tokens and update the dataframecell with mean vector

    read the coreference fields of the dataframe and create groups of mentions :
      Case 1: If it is a contiguous mention, combine the tokens and their embeddings into 1 group
      Case 2 : If it is a Singleton mention, leave its token and embeddding as it is
      Case 3: Discontiguous mention, read entire dataframe, find all the tokens that
              belong to the same chain for eg: 33a, 33bb, 33c etc. all belong to same
              identity chain
    create an interim dataframe with groups of mentions in above cases clubbed

    for every group of mention of any case (singleton, contiguous,discontiguous):
      find the mean vector of each group's contextual embeddings vector and store that in a final dataframe

    create a N x N Gold standard matrix where N is the no. of groups (no. of rows in final DF)
    for every row in the final dataframe:
      if that group's mention number is same:
        Gold standard matrix cell value = 1, 0 otherwise

    for threshold values in range of (0.75 to 1.0):
      create a N x N all pairs cosine similarity matrix, where N is the no. of groups (no. of rows in final DF)
      find the cosine distance between every group's mean contextual embedding vector
      and store it in above matrix
      if that cell's value is > threshold:
        update the above matrix value =1,0 otherwise

      flatten out both Gold standard (actual) and all pairs cosine similarity (Predicted) matrices
      create a confusion matrix

      calculate True Positives, False Positives, True Negatives & False Negatives per model per file per threshold
    calculate the combined F1 scores per model per threshold value

Draw a line chart to plot the performances of each model on same corpus
```

Figure 9: Calculating contextual word embeddings per model, finding similar words using different cosine similarity thresholds and comparing the results with golden standards to calculate F1 scores per model with different thresholds

The main project code can be found at :

https://colab.research.google.com/drive/1i13nTa7oWeXT8-K3mJwEISOYJOwXjhhv?usp=sharing

3.4 Challenges Faced

There were many challenges processing these full text documents and specially with the given file formats. The primary challenge was understanding the raw data, the purpose of the task, the models to be selected, pre-processing the raw data from knowtator to CoNLL format. Further, the main challenge was the size of files being huge, the steps of tokenizing and extracting token embeddings, creating huge adjacency matrices for five different pre-trained models and then calculating the F1 scores consumed huge amount of time.

We could also run the dataset on all 5 different models on 5 separate notebooks taking a decentralized approach but, we will have to import the final F1 scores to plot the final graph of F1 scores vs the threshold.

Chapter 4 Evaluation & results

4.1 Results

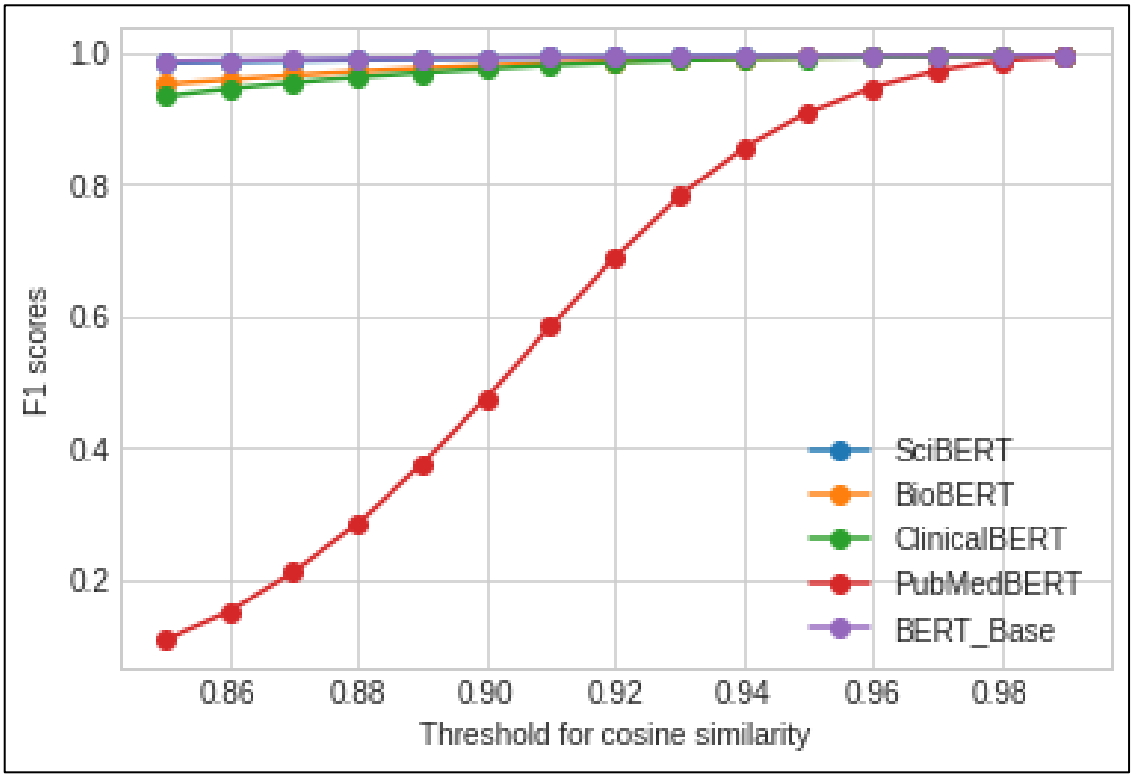

Figure 10. Threshold vs F1 scores line graph for threshold ranging from 0.85 to 1.00

In Figure 10. we can see a line graph plotted that show the performance of each model for various threshold values used for Cosine similarity between the groups that contain the mentions and then finally the F1 scores computed for every file at a single threshold value against the gold standard mentions. The results are for all the 40 files taken into consideration with their full-length text documents. This graph shows some very odd behaviour which we will discuss in the next unit.

4.2 Graphical Representation

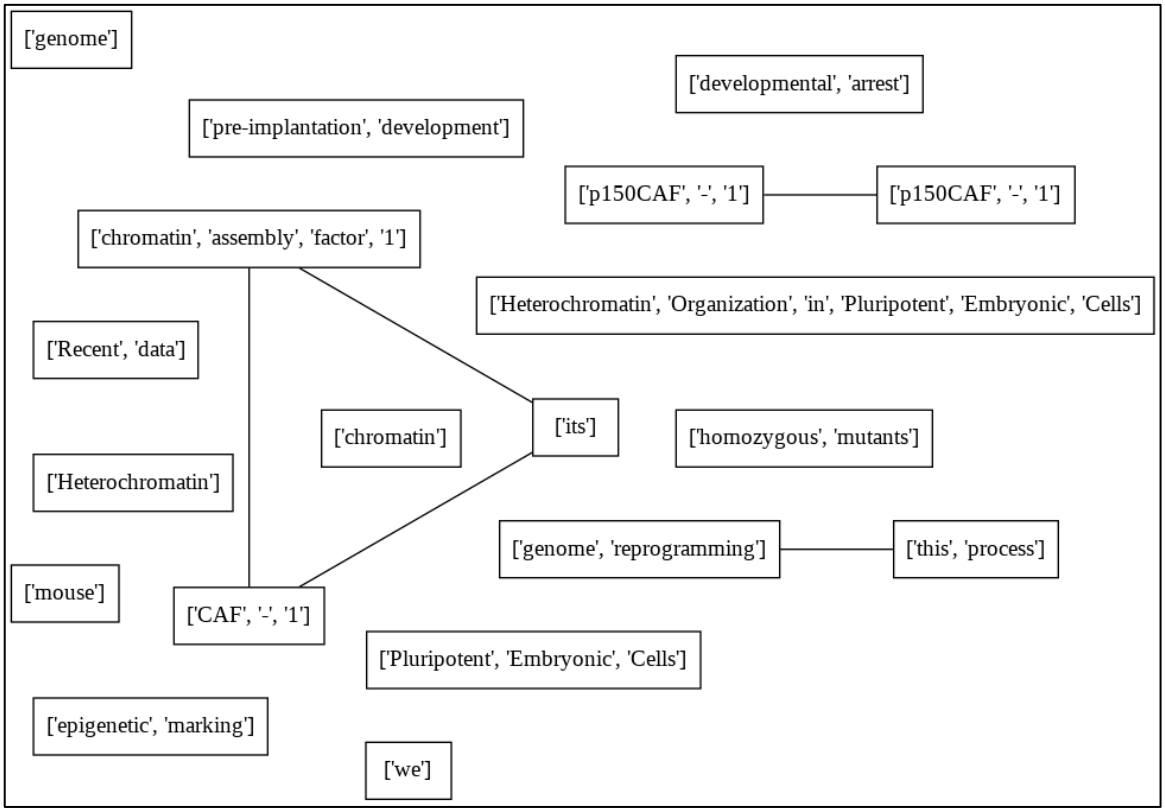

Figure 11. The relations between the Gold Standard (Actual) mentions from a sample extracted from document no. 17083276 [17]

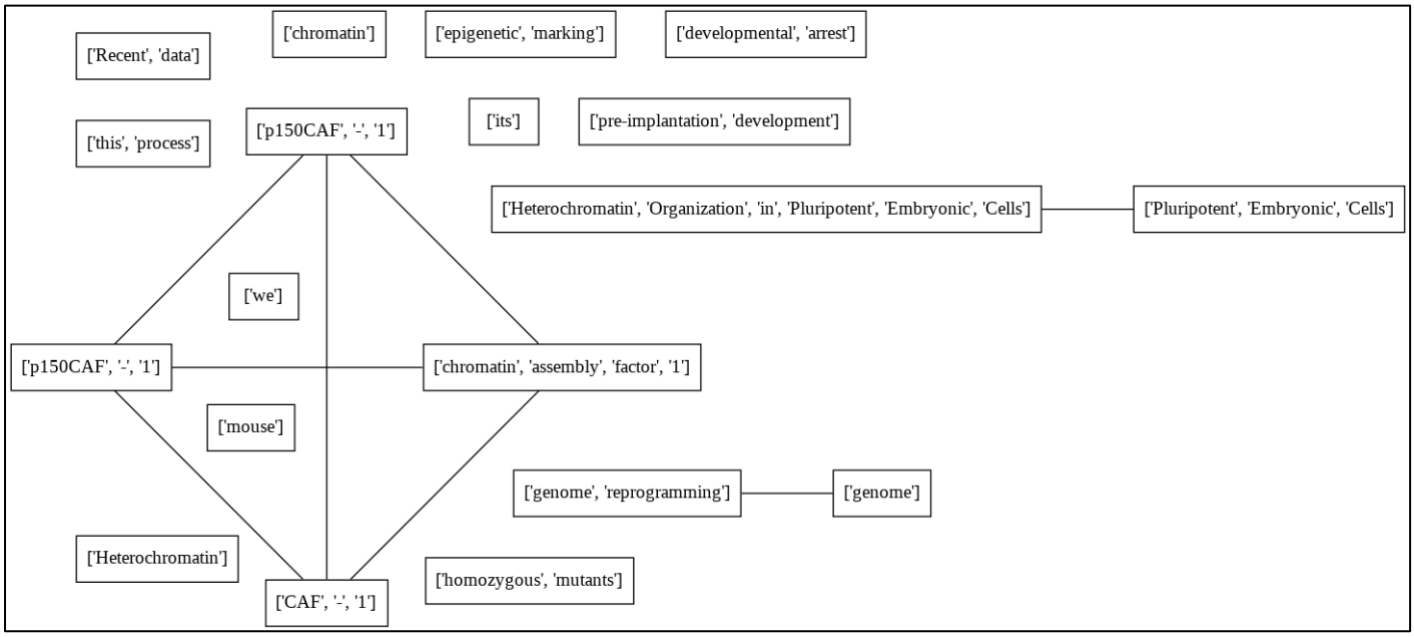

Figure 12. The relations between the predicted mention groupings from sample extracted from same document no. 17083276 [17]

Here, we have used Pygraphviz to automatically generate a graphical representation of the mentions that have been deemed to be linked together by "gold standards" i.e. the CRAFT group have annotated these mentions to be linked together in Figure 11, for document number 17083276 in the test files repository. In Figure 12 we have shown a similar graph of what PubMedBERT has predicted the mentions that seem to be linked i.e. which mean-token-embeddings have less difference between each other using cosine distance value less than the threshold value of 0.99 as fixed by the training task, for same document number 17083276 in the test files repository. Since the Pygraphviz consumes lots of resources and time, for which we had some constraints, we decided to use only a sample of text from a single document for representational purposes only, which gave us some interesting point for discussion that will be explained in the subsequent unit.

4.3 Our Findings

- In unit 4.1 we closely examine the F1 scores vs the threshold marks line graph and notice that while models like BERT Base, BioBERT, SciBERT, ClinicalBERT perform almost in a very similar manner, the PubMedBERT behaves in a very different manner.
- While all other models have an almost flat line gain in F1 scores with increase in the threshold values, the PubMedBERT increases in a very exponential fashion to meet the F1 scores of all other models at a common threshold.
- This displays a very peculiar behaviour of the PubMedBERT model which is also an adaptation of the BERT base model just like the other 3 models are also the adaptation of the BERT Base model.
- In section 4.2, we closely observe the graphical representations of 1. Gold standard mentions that are deemed to be representing the same entity by the CRAFT corpus in Figure 11. Here the words that belong to the same numerical value of the mentions are linked together via an adjacency matrix.
- In the gold standard graph, we can see that the words [chromatin,assembly, factor,1] , ['its'] and [CAF,-,1] are all referring to the same entity in the original text.
- In Figure 12, we see the graphical representation of the group of mentions that are predicted to be similar using the PubMedBERT model's extracted token embeddings per word.
- In the predicted graph, we can see that the mentions [chromatin,assembly, factor,1], [p150CAF,-,1] and [CAF,-,1] are linked together to be representing the same entity.
- This can be very well because the words with different meanings have very low similarity in their context vectors and similar words have very high similarity semantically.

Chapter 5 Conclusion

In this section we conclude our study with some arguments, constructive criticism and also a vast scope of study which can be explored further, giving a new direction of research in any or one of the models.

5.1 Study conclusion

- From the results and observations made above, we have
- In this study we have noticed that with the increase in the corpus on which the models are being pretrained, their behaviour is quite similar in terms of distinguishing the tokens of every word that are in any domain, and extracting their corresponding contextual vectors.
- This contradicts some of the literature review done which claim that different models have very distinguished impacts on different corpus, based on whether they have been pretrained from scratch on any particular domain, or have undergone any kind of continual training i.e. learning from a general domain and then continuing to be trained on a specific domain.
- Apart from any and all the models, one specific model that stood out is the PubMedBERT model, which shows some unforeseen results and different behaviour from the BERT base model and its adaptations of other pre-trained models in their respective domains.
- This goes to show that PubMedBERT is trained on either very different domain than any other model.
- A prime conclusion can be derived is that maybe rest of the models have undergone continual training from some kind of general corpus, whereas the PubMedBERT has been trained from the scratch only on the PubMedCentral corpus.

5.2 Future work

This study opens a lot of scope in the research of the PubMedBERT model specially, to see how it works, what is the architecture of the model, how it is trained differently from other adaptations of the BERT model. We need to understand how it works, what is the internal construction of the model, and how is it different from others. Further we can also explore the feature of context vector generation and try inputting not just one sentence but larger chunks of text at a single time into the models to see if they capture more contextual information per token and if the coreference resolution is done better. A new direction of study can be probed into the impact of many more pre-trained models on a cross domain corpora.

Appendix A Dictionary for this study

Here we will explain few terminologies that have been used in this report very frequently.

Word	Definition
Anaphora	Using terms like pronouns to refer a word/noun that has been used before in the text eg: he,she, his, her, they, it, its etc.
Antecedent	A term that gives meaning to its proform. Eg: Smit is fit because he goes to the gym. Here "he" refers to "Smit" therefore Smit is an antecedent of he.
BERT	Bidirectional Encoder Representation from Transformers – all encoders based deep learning model pretrained on Wikipedia and BookCorpus introduced by Google in 2018.
CoNLL	Coreference on Conputational Natural Language Learning – a yearly conference organized by SIGNLL which is an SIG focued on NLP. They have a separate file format used for Coreference resolution as described in Figure 7
Coreference Resolution	"Task of finding expressions that belong to same entity in text" [16]
Discontinuous	A mention that has spans spread across the document but are not sequential
Entity	Entities or named entities are the specific texts that belong to pre-defined categories, usually proper nouns. These are usually antecedents in the NLP domain
Mention	A mention is a reference of an entity that has appeared in texts. Mentions are nothing but Anaphora used in Natural Language Processing
Singleton	A mention that has span of just one element
Span	A span is a slice of a text document that contains a sequence of strings

Bibliography

[1] https://www.sas.com/en_gb/insights/analytics/what-is-natural-language-processing-nlp.html

[2] Zheng J, Chapman WW, Crowley RS, Savova GK. Coreference resolution: A review of general methodologies and applications in the clinical domain. Journal of biomedical informatics. 2011 Dec 1;44(6):1113-22.

[3] Lu P, Poesio M. Coreference Resolution for the Biomedical Domain: A Survey. arXiv preprint arXiv:2109.12424. 2021 Sep 25.

[4] Ng V. Entity Coreference Resolution. IEEE Intell. Informatics Bull.. 2016;17(1):7-13.

[5] Lee K, He L, Lewis M, Zettlemoyer L. End-to-end neural coreference resolution. arXiv preprint arXiv:1707.07045. 2017 Jul 21

[6] Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020 Feb 15;36(4):1234-40.

[7] lsentzer E, Murphy JR, Boag W, Weng WH, Jin D, Naumann T, McDermott M. Publicly available clinical BERT embeddings. arXiv preprint arXiv:1904.03323. 2019 Apr 6.

[8] Beltagy I, Lo K, Cohan A. Scibert: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676. 2019 Mar 26.

[9] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018 Oct 11.

[10] Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, Naumann T, Gao J, Poon H. Domain-specific language model pretraining for biomedical natural language processing. ACM Transactions on Computing for Healthcare (HEALTH). 2021 Oct 15;3(1):1-23.

[11] Nguyen N, Kim JD, Tsujii JI. Overview of the protein coreference task in BioNLP shared task 2011. InProceedings of the BioNLP Shared Task 2011 Workshop 2011 Jun 24 (pp. 74-82).

[12] D'Souza J, Ng V. Anaphora resolution in biomedical literature: a hybrid approach. InProceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine 2012 Oct 7 (pp. 113-122).

[13] Clark K, Manning CD. Improving coreference resolution by learning entity-level distributed representations. arXiv preprint arXiv:1606.01323. 2016 Jun 4.

[14] Trieu HL, Nguyen AK, Nguyen N, Miwa M, Takamura H, Ananiadou S. Coreference resolution in full text articles with bert and syntax-based mention filtering. InProceedings of The 5th Workshop on BioNLP Open Shared Tasks 2019 Nov (pp. 196-205).

[15] https://towardsdatascience.com/3-types-of-contextualized-word-embeddings-from-bert-using-transfer-learning-81fcefe3fe6d

[16] https://nlp.stanford.edu/projects/coref.shtml

[17] Cohen KB, Lanfranchi A, Choi MJ, Bada M, Baumgartner WA, Panteleyeva N, Verspoor K, Palmer M, Hunter LE. Coreference annotation and resolution in the Colorado Richly Annotated Full Text (CRAFT) corpus of biomedical journal articles. BMC bioinformatics. 2017 Dec;18(1):1-4.

[18] Ogren PV. Knowtator: A plug-in for creating training and evaluation data sets for Biomedical Natural Language systems. InProceedings of the 9th International Protégé Conference 2006 Jul 24 (pp. 73-76).

[19] https://sites.google.com/view/craft-shared-task-2019/craft-cr#h.p_z7fjwcIXrmkO

[20] https://gist.github.com/jakelever/d0539c2e9aead8a579f952aec9f8f563

[21] https://github.com/UCDenver-ccp/craft-shared-tasks/wiki/Evaluation-via-Local-Installation

[22] https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b