Disambiguating abbreviations through Negative Sampling

Mohammadreza Hosseini, Parvaneh Soleimanybaraijany, Francesco Zangrillo and **Enab Muneer**
Master's Degree in Artificial Intelligence, University of Bologna
{ mohammadrez.hossein2, p.soleimanybaraijny, francesco.zangrillo2, enab.enabmuneer}@studio.unibo.it

Abstract

The purpose of this experiment is to demonstrate the methodologies used to tackle an NLP task focusing on Abbreviation Disambiguation (AD) in the medical field. The goal of the abbreviation disambiguation task is to associate abbreviations in sentences with their corresponding full forms among a set of potential expansions. Leveraging state-of-the-art NLP techniques, a comprehensive system was created to accurately interpret abbreviations based on their contextual information through a negative sampling approach. Our study involved fine-tuning three pre-trained models: TinyBert, BioBert and SciBert, alongside a baseline. The results indicated that SciBert, the largest model among them, outperformed the others with an F1-score of 0.81. In comparison, TinyBert achieved F1-score of 0.30, the baseline model achieved 0.34, and BioBert achieved 0.64.

1 Introduction

Abbreviation disambiguation in the medical field refers to the process of resolving the full form of abbreviations commonly used in medical literature, clinical documentation, and healthcare settings. The medical field is notorious for its extensive use of abbreviations, which can often lead to confusion and miscommunication due to the potential for multiple meanings for a given abbreviation. Since abbreviations can vary across different medical specialties, regions, and even healthcare institutions, it is crucial to accurately interpret their intended meanings to avoid errors that could impact patient care and safety. For instance, with regard to the abbreviation "PCP," there exist three distinct expansions, namely "pneumocystis carinii pneumonia," "primary care provider," and "pentachlorophenol". Within the present dataset, the number of expansions for each abbreviation fluctuates within a range spanning from 1 to 65, from which we have to either ignore abbreviation with only one possible expansion or to find more examples which include other expansions as well. In addition, having knowledge about the connection between abbreviation and their expanded forms is advantageous for various tasks in natural language processing, such as question answering and machine reading comprehension.

To address this issue, various resources have been developed to aid healthcare professionals in distinction of medical abbreviations, such as medical dictionaries, electronic health record systems with built-in abbreviation expanders, and standardization efforts by professional organizations. Besides, In the field of artificial intelligence, numerous endeavors have been dedicated to addressing this task. Typically, when dealing with abbreviation disambiguation by deep learning models, it is commonly approached as a task of sequence classification (Veyseh et al., 2020). The objective is to match the provided abbreviation within its context to the appropriate expansion from a dictionary of potential expansions. Furthermore, the token classification approach is utilized as well in the context of abbreviation disambiguation (Myers et al., 2022). This method involves assigning specific labels to individual tokens within the given acronym in order to identify the corresponding expansion from a candidate expansion dictionary.

We adopted the architecture proposed in (Wu et al., 2022) as the basis for our research, making minor modifications to suit our specific objectives. The implemented framework uses prompts, along with a unique strategy for negative sampling for abbreviation disambiguation. Our approach involves several steps. Firstly, we create different prompt templates and use them to combine the context of the abbreviation with potential expansions. Next, we employ a pre-trained language model like BERT to encode the combined context separately. A linear layer is then used to convert the context vectors into logits. Given that the number of candidate

expansions can vary for each abbreviation, we aim to generate negative samples by randomly padding the expansions. Lastly, we distinguish between the original negative expansions (considered hard negative samples) and the added ones (considered easy negative samples). This distinction allows us to calculate an additional loss, resulting in a more robust system.

2 Background

Due to the adaptation of a prompt-based model and utilization of negetive sampling approach in this study, this section aims to provide an overview of these two approaches and their key characteristics. Negative sampling is a technique used in machine learning, particularly in the context of training models for tasks such as word embeddings or recommendation systems. In many machine learning tasks, the training data consists of positive examples (instances that should be predicted correctly) and negative examples (instances that should be predicted incorrectly). Instead of considering all possible negative examples during training, negative sampling randomly selects a small subset of negative examples to be included in each training batch. The idea behind negative sampling is that by focusing on a small set of negatives, the model can learn more efficiently and effectively discriminate between positive and negative examples.

A prompt-based algorithm in Natural Language Processing refers to a technique where a model is designed to generate responses based on a given prompt like a query or specific instruction. In a prompt-based algorithm, the input to the model consists of a text prompt that provides initial context or instructions for generating the desired output. The model then processes the prompt and generates a response based on its understanding of the language and the patterns it has learned during training.

3 System description

Our project's problem statement entails disambiguating the expansion of a desired abbreviation within an input context. The context is formed using words and abbreviations (W: w1, w2, ...) and the position of desired abbreviation is also given. We aim to determine the correct expansion from a pool of N possible candidates.

The cornerstone of the implemented architecture lies in the input provided to the model. Instead of solely transmitting tokenized contexts to the model, our approach involves transmitting a combination of the text and an expansion related to the abbreviation which is selected from the list of candidates for that abbreviation. Furthermore, The original model proposed by (Wu et al., 2022) added a prompt to the input of the model that provide additional information to the language model (see figure 1 for the detail of the model). We made an attempt to incorporate their approach of employing prompts in our model. However, this implementation did not yield any notable enhancements in the model's performance when compared to using negative sampling alone. It appears that the concept of prompt-based modeling does not align well with our specific problem and the characteristics of our dataset.

The candidate list consists of two parts: original candidates and additional negative candidates. The original candidates are predetermined for each abbreviation and their sizes can vary. We have stored the original candidates (possible labels) in a dictionary, with each abbreviation having its own set. On the other hand, the additional negative candidates are strictly outside of the original set which is an improvement in comparison to the base paper (negative sampling). The significance of this meticulous choice lies in its impact on the hinge loss, which will be disscussed in depth at a later stage. To create a combined list of candidates with length K+2, we concatenate these two sets. K is determined by the maximum number of original expansions for any abbreviation (which for the current selected data it is equal to 23) and 2 is an additional length in order to guarantee negative sampling (in the situation the abbreviation has the maximum number of available expansions) . For instance, if the number of original candidate expansions is 21, we need to choose 4 additional negative expansions to reach the max length.

In order to do some experiments considering prompt-based apprach, we created two prompts to enrich the input and provide essential context and guidance for the language model, each serving a distinct purpose. The first prompt we utilized is based on the original paper that inspired our research. we introduced an additional prompt.

Here you can find the detail of each prompt:

- First prompt: 'the meaning of *abbreviation* is or equals *expansion*'

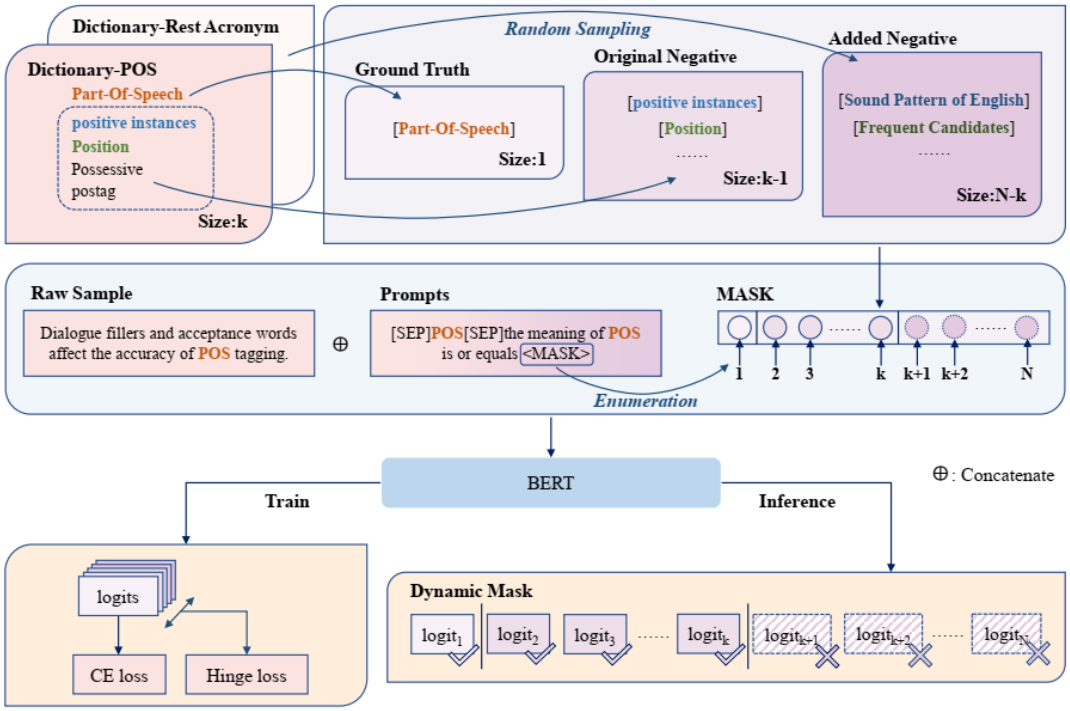

Figure 1: Model architecture poposed by (Wu et al., 2022)

- ABV prompt: *abbreviation* and *label* both include the following letters: *letter1* and *letter2* and *letter3*

The task of abbreviation disambiguation is treated as a classification problem, where the classes are the expansions (the list of candidates for each abbreviation). The model structure consists of a BERT-based architecture with an additional linear layer. The BERT model is loaded using the *AutoModel.from_pretrained* method and serves as the main component of the model. It encodes the input sequence and produces a pooled output. The additional linear layer, applies a linear transformation to the pooled output, mapping it to a single-dimensional output tensor. This layer helps to capture further patterns and relationships within the encoded representation. Overall, the model combines the power of BERT's contextualized embeddings with a linear layer to generate the final output of our tasks.

Defining a specialized loss function is a crucial aspect of our model's structure, as it enables us to address the specific requirements of our project effectively. In our endeavor, we have employed a special loss function tailored to address the specific requirements of our problem. Our loss function is designed to achieve two primary objectives for our model. Firstly, we aim to ensure that the ground truth expansion, which represents the desired outcome, receives the highest score among all the potential expansions. This objective is crucial as it ensures the model's ability to accurately prioritize the most appropriate expansion for a given input. To accomplish this, we have utilized the cross entropy loss function, which measures the dissimilarity between predicted and ground truth expansions, effectively guiding the model towards assigning the highest score to the correct expansion. Secondly, we also strive to ensure that the original negative expansions, which are initially provided as part of the training data, receive higher scores than additional negative expansions. By incorporating these goals into our specialized loss function, we aim to enhance the performance and reliability of our model in generating accurate and contextually appropriate expansions for a given input.

Our specialized loss function is inspired by the concept of hinge loss. This goal focuses on maintaining a margin between the scores of the original negative expansions and additional negative expansions. To achieve this, we define the original expansion scores as S_{ori}= {$score_1$, $score_2$, ..., $score_k$} and the additional expansion scores as S_{add} = {$score_{k+1}$, $score_{k+2}$, ..., $score_N$}. We aim to ensure that the

minimum score among the original expansions is higher than the maximum score among the additional expansions by a margin called *gap*. This margin is then utilized in the hinge loss formulation as follows:

$$Loss_{\text{hing}} = max(gap - min(S_{\text{ori}}) + max(s_{\text{add}}), 0) \quad (1)$$

By examining the maximum score of additional negative expansions, we aim to identify instances where our model struggles to distinguish between original expansions and randomly selected negative examples. This score serves as an indicator of the model's weaknesses in detecting negative expansions. On the other hand, analyzing the minimum score of the original expansion allows us to measure the model's accuracy in recognizing the intended expansions. A lower score suggests that our model exhibits a reduced level of error and is more successful at identifying the original expansions correctly.

Finally, our overall loss function combines the cross entropy loss with a weighted hinge loss:

$$Loss = Loss_{\text{CE}} + mu * Loss_{\text{hing}} \quad (2)$$

Where *mu* is a hyperparameter controlling the ratio of the hinge loss component. By integrating this specialized loss, we effectively guide our model to maintain the desired margin between the scores of original and additional negative expansions, contributing to the reliability and robustness of the generated expansions.

About ths baseline model, it utilizes a scoring mechanism to evaluate the words in potential expansions. This involves counting the occurrences of each word within the context and summing them together. The resulting sum is then divided by the total number of words in the expansion. The expansion with the highest score is regarded as the correct choice. In the event of multiple expansions having equal scores, the first expansion in the list is selected as the default option. By relying on contextual word frequency, this methodology aims to identify the most suitable expansion for a given context.

4 Data

In this expermint we utilized **Me**dical **D**ataset for **A**bbreviation Disambiguation for Natural **L**anguage Understanding (MeDAL) created by (Wen et al., 2020). The MeDAL dataset basicly consists of 14,393,619 articles, considering the huge size of the dataset and the associated computational cost, a subset of 5 million data points are sampled from the complete corpus and is available to download, which are split into 3 million training samples, 1 million validation samples and 1 million test samples. Every entry in the dataset comprises several elements, namely an Id, a textual content, the position of abbreviations within the text, and a label representing the complete form of the abbreviations. Furthermore, we have introduced a new column called 'ABV' to the data, which contains the abbreviations themselves. An essential aspect concerning the MeDAL dataset is its utilization of reverse substitution for generating samples without human labeling. This approach increases the likelihood of making errors during the automatic labeling process. Later on, we will delve into the discussion of some errors discovered within the dataset.

By grouping the abbreviations together, we can analyze the data more effectively. This analysis reveals a total of 5798 distinct abbreviations, each with varying numbers of supports in the dataset. The abbreviation that has the lowest number of support is represented by 10 instances in the dataset, whereas the abbreviation that enjoys the highest number of support is represented by a significantly larger count of 23774 occurrences.

In order to effectively handle our extensive joint dataset of 5 million records, we have made the decision to work with a reduced fraction of the data. As we wanted to preserve the complexity of the data, we employ a strategy of excluding rows containing abbreviation with low number of expansions, i.e., those with fewer than 20 expansions, as they tend to be less complex for the model. Simultaneously, due to limitations in the available resource for processing vast amounts of data, we also eliminate rows containing abbreviation with high number of expansions, i.e., more than 30 expansions. This subset size has been chosen to strike a balance between demonstrating the objectives of our project and ensuring feasibility within our available resources. Furthermore, we implemented a data filtering process to extract specific samples for each expansion. Specifically, we selected a total of 30 samples for each expansion.

After taking into account the minimum and maximum borders, we found that our dataset remained quite large. Therefore, we randomly chose only

four abbreviations from the remaining data. Subsequently, we examined the final chosen data and noticed that these entries contained abbreviations with a range of expansions between 20 and 23. Once the aforementioned selections were completed on the entire available data, we proceeded to divide it into training, validation, and test sets. The training set is consisting of 65 percent of the selected data, while both the validation and test sets, each contains 17.5 percent. To perform the data split, we utilized stratified sampling, which ensures that the distribution of possible expansions is preserved across all three splits. Additionally, we assume all the possible expansions of the abbreviations are presented in the dataset in hand, Which means our model would not detect correct expansions solely from the abbreviation but it chooses among a list of known expansions.

In practical scenarios, it is possible to encounter unseen abbreviations. However, since our primary focus revolves around managing abbreviations within our dataset, this concern does not carry significant weight. Our proposed model does not aim to comprehend the broader concept of abbreviations, but rather seeks to accurately identify and interpret the specific abbreviations relevant to our dataset and only among the proposed expansions.

Preprocessing is a critical step in NLP tasks as it helps to improve the accuracy of models by removing irrelevant information, standardizing the text format, correcting errors, and making it easier for deep learning models to identify patterns and make predictions. To prepare our data to be ready for further manipulation we did some preprocessing on it including label correction. Label correction becomes crucial in this task as the primary objective is to identify the accurate relationship between an abbreviation and its expanded label. It is essential to provide the model with the accurate labels and informative data in order to achieve this goal. For this reason, we tried to improve the consistency and accuracy of labels in the dataset by removing stop words, aligning words of the labels with the 'ABV' value, and providing corrected labels. We provided a mechanism to automate the process of label correction based on the specific requirements and characteristics of the dataset.

The initial step of our data processing involves the segmentation of the data based on the column labeled 'ABV'. This segmentation creates distinct groups for each unique value found in the 'ABV' column. We proceed by identifying the unique labels associated with each 'ABV' value and eliminating any stop words present in these labels. We thought about removing stop words from the labels because sometimes stop words are included in the label to ensure grammatical correctness, but they don't actually contribute to the abbreviation. This can make the training process challenging for the model. Subsequently, we verify if any modifications have been made to the labels through the removal of stop words. If such modifications are detected, a mapping dictionary is generated. This dictionary establishes a connection between the original label and its modified version.

Additionally, we identify "odd" labels that consist of more words than the abbreviation they are associated with. In order to tackle this problem, we make an effort to rectify these labels by aligning their words with the abbreviation. This involves comparing each word in the label with the corresponding letter in the abbreviation. We don't simply consider the word that directly corresponds to the letter in its exact order, but we also attempt to find the relevant word by considering even one word before and after the expected position. Finally, by selecting matching or partially matching words, we construct a corrected label. These corrected labels are then added to the mapping dictionary to check later on. While reviewing the modified labels, we encountered certain exceptions. Specifically, we observed that some abbreviations include a numerical digit, and the corresponding word in the expanded label is the ordinal version of that number (for example, the abbreviation may contain the digit "4," while the corresponding word in the label is "fourth"). We decided to exclude these labels from the correction process.

After completing the mapping dictionary, we conducted a manual check and discovered certain instances of data issue. As an example, consider the abbreviation 'AD' and one its labels in the data, which is 'dementia of the Alzheimer type' (whereas the correct abbreviation of this label is 'DAT' not 'AD' which proves the existence of errors in data). Our correction algorithm assumes that the label associated with this abbreviation should consist of two words starting with 'A' and 'D' respectively, in that specific order. However, the provided label does not meet the criteria for correctness, leading to its modification to a label containing only 'Alzheimer' which is not correct.

Furthermore, we encountered additional exceptions in the labels that required manual correction. For example, the label "modified vaccinia virus ankara" associated with the abbreviation 'MVA' was not classified as an incorrect label by our algorithm, even though it requires correction. This is because there are two words starting with 'V' in the middle of the label, which both match with the 'V' in the second position of 'MVA'. We manually included this specific case in the mapping dictionary to address such instances.

In certain instances, we have observed peculiar patterns in the potential expansions of abbreviations. Take, for instance, the abbreviation 'CR', which could be expanded as 'complete remission rate', 'complete regression', 'complete response rate', or simply 'complete'. Here, an evident issue arises when 'complete' is abbreviated as 'CR', which is problematic. This issue stems from an overlooked word beginning with 'R' following the word 'complete'. Resolving such cases requires the expertise of a human specialist. As a temporary measure, we have chosen to exclude these samples from consideration. Similar situations also arise with the abbreviation 'GA', where potential expansions include 'general', 'general anaesthetic', 'general anesthesia', and 'general anaesthesia'. This list is by no means exhaustive.

5 Experimental setup and results

All models are constructed using the Huggingface transformers library, an open-source framework known for its extensive range of pre-trained models (Wolf et al., 2020).

We performed fine-tuning on three pre-trained models using our own data: TinyBert, BioBert, and SciBert. Initially, we conducted various experiments on the TinyBert model using different prompts. These experiments involved testing the model with input that had no prompt, input containing the prompt suggested by the original paper, input containing our own proposed prompt, and a model that considered both prompts simultaneously. We chose TinyBert for these experiments because it is smaller in size compared to the other two models. As a result, it can be trained more quickly and requires fewer resources. The purpose of conducting these four experiments was to analyze the impact of prompts on our data and determine the optimal configuration for BoiBert and SciBert. Through the evaluation of various prompts on the TinyBert model, we discovered that our model achieves better performance when no prompt is included. Consequently, for the subsequent models (BoiBert and SciBert), we trained them without employing a prompt-based approach. Instead, we focused on incorporating negative sampling techniques to enhance their performance.

In the experiments, we selected the hyperparameters based on the values recommended in the original paper which are as follows:

- The optimizer is AdamW with learning rate as 3e-5.
- The batch size is equal to 2.
- We pad or cut the input into 128 length.
- The number of epoch is equal to 2.
- Gap and mu, two parameters related to loss fuction are equal to 0.2 and 0.5 respectively.

We employ the F1-score as our evaluation metric to assess the performance of our models. Table 1 and figure 2 present tabular and pictorial summary of the results obtained from the four experiments conducted on the TinyBERT model, respectively.

Prompt	Train	Validation	Test
No	0.31	0.28	**0.30**
First	0.30	0.26	0.26
ABV	0.29	0.25	0.26
All	0.25	0.21	0.19

Table 1: Comparison of F1-score on TinyBert model Considering different prompts

Table 2 and figure 3 present summary of the results achieved by all models when no prompt was added to the input. Based on the result analysis, it is evident that the SciBert model consistently outperformed the other models across all data splits. Both BioBert and SciBert exhibited superior results compared to TinyBert, primarily due to their larger size and more extensive language representation capabilities. Furthermore, the specific nature of our dataset, MeDAL, which is derived from PubMed abstracts, played a role in contributing to the superior performance of SciBert compared to BioBert. PubMed is a search engine that indexes scientific publications in the biomedical field, indicating that SciBert's design aligns well with the specialized content of our dataset.

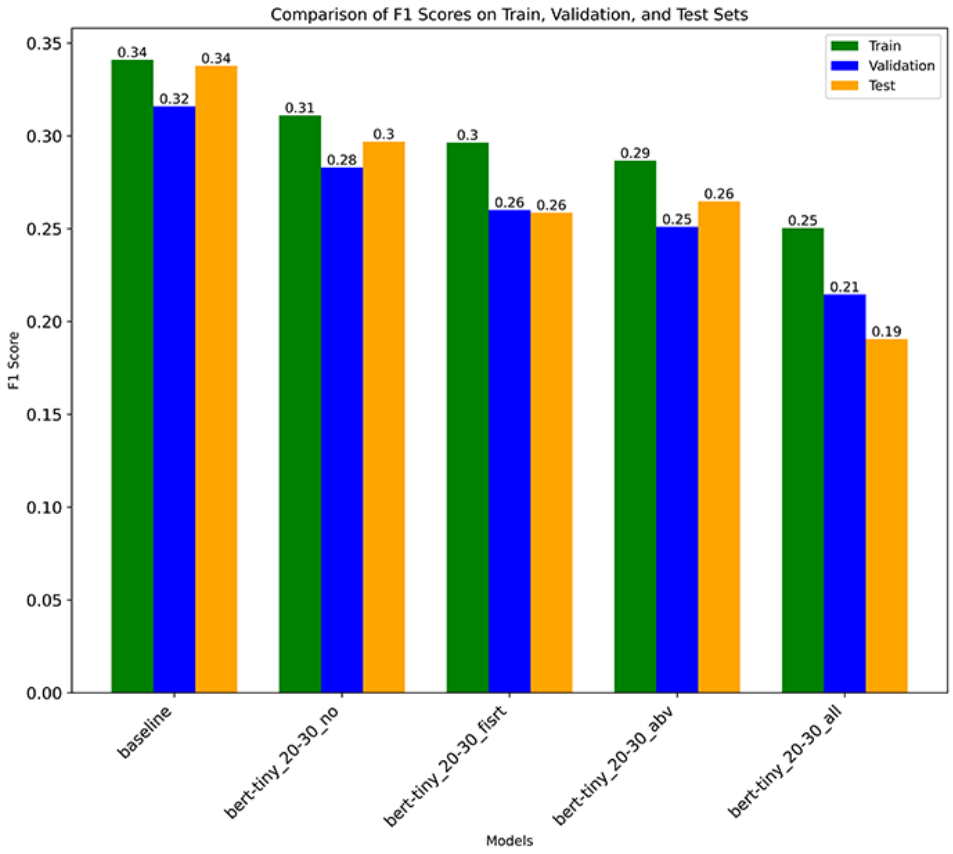

Figure 2: Comparision of F1-score related to TinyBert alonside of the baseline on train, validation and test set

Model	Train	Validation	Test
Baseline	0.34	0.32	0.34
TinyBert	0.31	0.28	0.30
BioBert	0.74	0.62	0.64
SciBert	**0.88**	**0.79**	**0.81**

Table 2: Comparison of F1-score on all models (with no prompt)

Furthermore, to examine the four abbreviations in our data, we generated classification reports. See table 3 for more details.

Label	Support	Precision	Recall	F1
AD	127	0.82	0.79	0.78
AH	119	0.93	0.90	0.89
AM	117	0.81	0.78	0.77
BA	101	0.85	0.82	0.80

Table 3: Comparison of all metrics on four abbreviations

6 Discussion

The results obtained demonstrate great promise and highlight the considerable potential of the implemented approach in achieving favorable outcomes. To identify specific types of errors made by the SciBert model (as the best model), we conducted an error analysis and examined the confusion matrix which provides a breakdown of the model's predictions compared to the actual labels or ground truth. (see figure 4).

Regarding the error observed in our model's misclassification of full forms with conceptually similar meanings, one of the primary contributing factors is the limited training data. The model lacks exposure to an ample and diverse range of training examples where abbreviations are misclassified, which would enable it to learn the distinctive characteristics of such cases. Due to this limitation, the model may struggle to accurately differentiate between full forms that share close conceptual meanings.

Here there is a list of miss-classified labels:

- The label **adult** in the dataset has been identified as incorrect and needs to be modified to either 'adult height' or 'adultdirected'. This

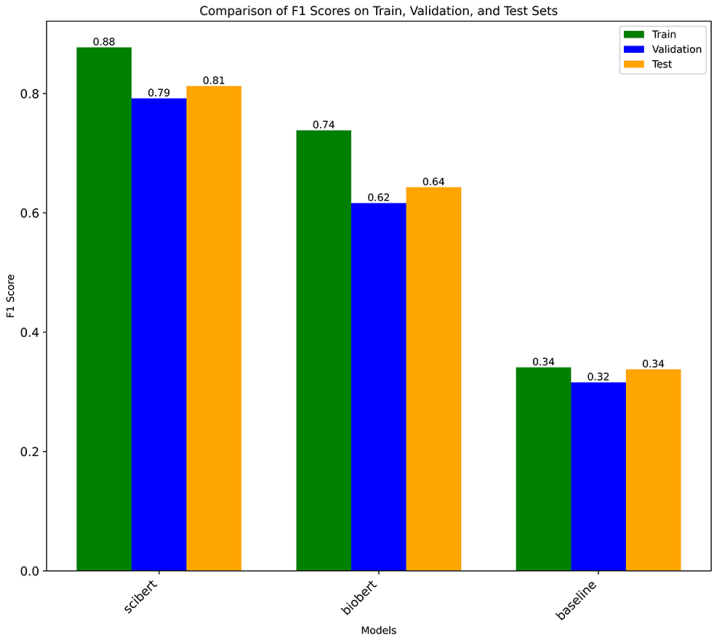

Figure 3: Comparision of F1-score for BioBert, SciBert (the models with no prompt) and baseline

correction requires the intervention of an expert human supervisor and is related to the process of dataset creation.

- The label **benzamide** and **nbenzyladenine** share some structural similarities, particularly the presence of a **benzene ring**. The model may be focusing on these common features and disregarding the specific functional groups that differentiate the two compounds. This could lead to misclassification.
- The label **amplitude modulated** is classified as **sinusoidally amplitudemodulated** which is completely correct. But due to the issues in the dataset it is considered as wrong. We may handle it in the postprocessing step by removing the word **sinusoidally** from them labels as it doesn't carry so much meaning, or we may normilize the labels to be either two-word labels or one-word labels. For example, changing all the label **amplitude modulated** to **amplitudemodulated** or vice versa.
- The label **paf acetylhydrolase** with the abbreviation **AH** is missclassified as **alveolar hemorrhage** which is showing the model has learned to pick a label within the original labels even when the true label is misleading. **paf acetylhydrolase** is also another label which has to be changed to **acetylhydrolase**, within the preprocessing step.
- The label **arbuscular mycorrhizal** refers to the type of symbiotic association between plants and fungi, whereas **arbuscular mycorrhiza** refers to the physical structure formed by the plant roots and the arbuscular mycorrhizal fungi. It can be handled in postprocessing. The same problem happend for label **acetoxymethyl** which classified as **acetoxymethyl ester** for three times.
- The label **anteromedial nucleus** with the full form **AM** is misclassified as **amygdaloid**. It could be handled by removing **nucleus** from the label in the preprocessing step.
- The label **anteromedial** is classified as **anteromedial nucleus**, which is incorrect and could be handled within the postprocessing by removing **nucleus** from the label.

The place in which the model has problem the most is the following: 'butyl acrylate' label corresponding to the abbreviation 'BA' is classified as 'nbutyl acrylate' for 3 times. The label 'adenomatous hyperplasia' corresponding to the abbreviation 'AH' is classified as 'atypical hyperplasia' for 4 times.

The model faces difficulties primarily in two cases. When encountering instances that are essentially the same thing but expressed differently, the model requires preprocessing to align and normalize these variations. When dealing with very

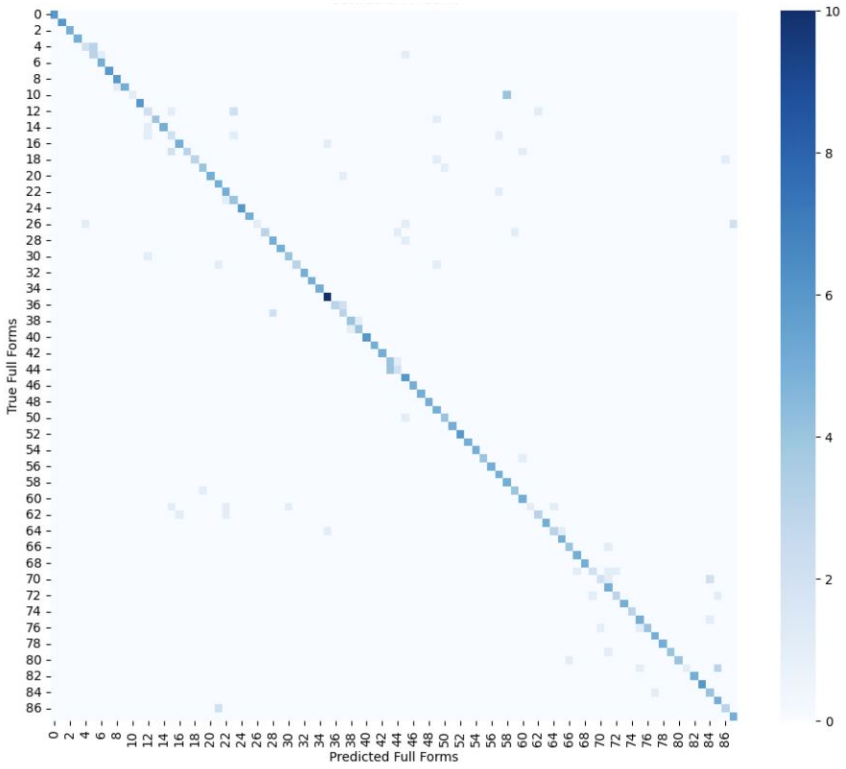

Figure 4: Confusion matrix on the result of SciBert model

similar concepts that the model struggles to distinguish, it indicates a need for more training samples to improve its ability to differentiate between such nuances.

Below, we provide a list of abbreviations and their corresponding full forms that are prone to presenting problematic and potentially inaccurate information. These findings were also obtained through an automated process, which could be further addressed in future work by involving human specialists:

- alternating: AC, This abbreviation could refer to alternating current, which is the flow of electric charge that periodically reverses direction. It could also stand for alternating copolymers, which are polymers composed of two or more types of repeating units that alternate along the polymer chain.

- alveolar: AM, This abbreviation could refer to alveolar macrophages, which are immune cells located in the alveoli of the lungs.

- abasic: AP, This abbreviation could refer to apurinic/apyrimidinic sites, which are locations in DNA where a nucleotide is missing.

- active: AS, This abbreviation could have a variety of meanings depending on the context. For example, it could refer to active transport, which is the movement of molecules across a membrane against a concentration gradient using energy. It could also stand for active site, which is the region of an enzyme where substrate molecules bind and chemical reactions occur.

- stress: AS, This abbreviation could refer to stress hormones, such as cortisol, that are released by the body in response to stress. It could also refer to stress testing, which is a diagnostic tool used to evaluate how well the heart responds to stress.

- assay: CA, This abbreviation could refer to a variety of types of assays, which are laboratory tests used to measure the presence, amount, or activity of a substance or biomolecule. For example, it could refer to enzyme-linked immunosorbent assay (ELISA), which is a common method used to detect the presence of antibodies or antigens.

After analyzing the errors, we identified recurring difficulties in the model's predictions. To

address these errors, we implemented a post-processing step called mapping. This mapping, called 'bad_label2good_label', resolves certain misclassifications by mapping incorrect labels to the correct ones. By applying the mapping code, which replaces incorrect labels with the corresponding correct labels, we obtained the final 'y_true' and 'y_pred' values for evaluation. Specifically, we replaced labels in 'best_model['y_true']' and 'best_model['y_pred']' if they matched the keys in 'bad_label2good_label'.

Subsequently, we performed a separate classification report specifically for samples with the 'AM' abbreviation. Since the 'bad_label2good_label' mapping primarily corrects errors related to this abbreviation, we focused on evaluating its impact. The results of this post-processing step revealed a significant improvement, with the 'f1 score' for 'AM' abbreviation samples increasing from 77% to an impressive 89%. This improvement highlights the effectiveness of the post-processing step in enhancing the model's performance.

7 Conclusion

In conclusion, this study aimed to address the challenge of Abbreviation Disambiguation (AD) in the medical field using state-of-the-art NLP techniques. The results demonstrate the effectiveness of leveraging contextual information and employing a negative sampling approach to accurately interpret abbreviations in sentences. By fine-tuning three pre-trained models (TinyBert, BioBert, and SciBert), a comprehensive system was developed for the AD task. Among the models evaluated, SciBert, the largest model, achieved the highest performance, with an impressive F1-score of 0.81. This indicates that SciBert successfully associated abbreviations with their corresponding full forms in the medical domain. These findings suggest that the choice of the pre-trained model plays a crucial role in achieving accurate abbreviation disambiguation. While smaller models like TinyBert and the baseline model may be more computationally efficient, they lack the capacity to capture the complex contextual nuances necessary for robust disambiguation.

The MeDAL dataset presents valuable information that has been extracted through an automated process. However, it is not without its challenges, some of which can be addressed through meticulous dataset inspection and additional automated techniques we have implemented in our code. Nevertheless, certain issues necessitate the expertise of human inspectors. To streamline the effort involved, it is crucial to evaluate and prioritize the most critical areas, eliminating unnecessary and redundant expansions of full forms. Only through this approach can we effectively handle the diverse range of possible expansions within our dataset.

Additionally, it is important to acknowledge our limited resources, which have impacted the final results obtained using the proposed architecture. To improve upon this, future work should include training the model on a larger dataset, encompassing more than just four abbreviations. Furthermore, employing more robust models such as BIOGPT, which is three times larger than SCIBERT with a .bin file size exceeding 1.5 GB, would be essential.

References

Mitchell Myers, Mucahit Cevik, Sanaz Mohammad Jafari, and Savas Yildirim. 2022. Token classification for disambiguating medical abbreviations. *arXive*.

A. P. B. Veyseh, F. Dernoncourt, Q. H. Tran, and T. H. Nguyen. 2020. What does this acronym mean? introducing a new dataset for acronym identification and disambiguation. *Proceedings of the COLING 2020, International Committee on Computational Linguistics*, pages 3285–3301.

Zhi Wen, Xing Han Lu, and Siva Reddy. 2020. MeDAL: Medical abbreviation disambiguation dataset for natural language understanding pretraining. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 130–135, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Taiqiang Wu, Xingyu Bai, and Yujiu Yang. 2022. Prompt-based model for acronym disambiguation via negative sampling. *AAAI'22: Scientific Document Understanding, CEUR Workshop Proceedings*.