Master Thesis

Improving Medical Text Classifiers with Balanced Datasets

Murat Ertas

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by: Piek Th.J.M. Vossen
2^{nd} reader: Luís Guilherme de Passos Morgado da Costa

Submitted: June 30, 2024

Abstract

This thesis addresses the improvement of medical text classifiers by enhancing the quality and balance of both training and test datasets. Imbalanced data poses significant challenges to the accuracy and reliability of Natural Language Processing (NLP) models, particularly in the medical domain where rare conditions and minority classes are often underrepresented. To tackle these issues, this research explores targeted data augmentation techniques, specifically through active learning, to iteratively improve the training data. The study integrates multiple datasets from different medical specialties to create a more balanced and comprehensive test set. Key interventions include re-annotating mislabeled data and incorporating a negative category to better manage false positives and negatives. Experimental results demonstrate that these strategies noticeably enhance the model's performance, providing more accurate and reliable classifications of medical texts. This work contributes to the field by proposing practical methods for dataset balancing and model optimization, ultimately supporting healthcare professionals with more precise NLP tools.

Declaration of Authorship

I, Murat Ertas, declare that this thesis, titled *Improving Medical Text Classifiers with Balanced Datasets* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 30 June 2024

Signed:

Acknowledgments

I would like to express my deepest gratitude to my thesis supervisor, Dr. Piek Vossen, for his invaluable guidance and for being a great teacher throughout this thesis. His patience in explaining every detail I needed to know was instrumental in the completion of this work.

I am also grateful to Edwin Geleijn, my supervisor at AUMC, who mobilized all necessary resources to provide for my material needs from the first day of my internship. Edwin, along with Marike van der Leeden and Sabina van der Veen, also provided essential help with the annotations.

I would also like to thank Luís Passos Morgado da Costa and Antske Fokkens for everything I learned from their valuable classes.

Lastly, I want to thank my beloved wife, Nina, for believing in me, motivating me, and providing great support, making this journey possible.

List of Figures

Contents

Chapter 1

Introduction

Medical text classifiers are Natural Language Processing (NLP) tools designed to understand and categorize pieces of text within a medical context. The quality and balance of datasets play a crucial role in the performance of text classifiers. Models created with imbalanced datasets can be biased towards their dominant categories and if they contain inconsistencies in their data they can they can yield inaccurate or unreliable predictions.

This thesis is written in collaboration with A-PROOF project which aims to develop medical text classifiers capable of determining a patient's functioning level from free-text clinical notes written in Dutch. These models can be utilized to analyze extensive clinical data, providing insights into recovery patterns within specific patient populations.

This thesis aims to improve a medical text classifier used by A-PROOF project focusing on these three questions:

1. How can the quality and balance of test datasets be enhanced to improve measuring the performance of medical text classifiers?

2. What strategies can be employed to improve the quality and balance of training datasets for medical text classifiers?

3. How can the model's ability to handle false negatives (FN) and false positives (FP) be optimized?

1.1 Thesis Overview and Objective

The primary goal of this thesis is to improve the effectiveness and reliability of the medical text classifier created by Kim (2021). The research addresses three key issues:

1. **Improving Test Data Quality:** This involves correcting misannotations (Schramm, 2023), particularly for one category that has been annotated incorrectly, and addressing observed false-false positives by re-validating their gold standard values.

   (a) **INS Category Re-annotation**: Correcting annotation mistakes in INS category.

   (b) **False-false positives Re-validation**: Addressing false false positive errors in the evaluation as some of them were misannotated and supposed to be true positives. The gold value of these instances are re-validated.

2. **Improving Test Balancedness:** The performance evaluation of a text classifier is heavily influenced by the composition of its test data. An imbalanced test set can lead to misleading performance metrics, skewing the perceived effectiveness of the model. To address this, I aim to create a more balanced and representative test dataset. This involves combining separate datasets to ensure a more balanced distribution of classes, which will provide a clearer picture of the model's performance across different categories. The approach involves combining three smaller, separate datasets from similar domains (medical notes from different medical specializations) to create a more balanced and representative test dataset.

3. **Improving Model Balancedness to Address FN and FP:** To improve the model's ability to handle false negatives and false positives, targeted data augmentation through active learning will be adopted. This method focuses on iteratively selecting the most informative data points to improve the model's performance. Active learning is employed to selectively augment the training data. By iteratively selecting the most informative samples for annotation, the training set can be enhanced with instances that are likely to improve the model's performance, particularly on rare or difficult cases. This strategy ensures efficient use of annotation resources and focuses effort on data points that are expected to have the most significant impact on the model's accuracy and robustness.

1.2 Thesis Outline

The thesis will follow a structured approach, starting with an introduction to the problem and objectives, followed by detailed sections on methodology, experiments, results, and discussion. Each section will build upon the previous one, leading to a comprehensive understanding of the improvements made to the medical text classifier.

The thesis begins with the present chapter which is the introduction of overall project and goals of the research, followed by the background chapter (Chapter 2) where I provide a glimpse of relevant literature to the research. In Chapter 3 (Data and Annotations), I will demonstrate previous and recent data that is relevant to this research. Following this, in Chapter 4 (Methodology), I will describe my methodological choices by relying on the literature provided earlier. Then, I will move on to Experiments and Results (Chapter 5, where I implement the methodological preferences in various set of experiments. In this chapter I demonstrate the results of my hypothesized experiments. Following this, I will discuss the output of this research in a broader perpective in the Discussion (Chapter 6). In this chapter I will also provide a general analysis of the output and discuss the possible future work that this thesis may expand into.

1.3 Research Contributions

This research contributes to the field by:

1. Demonstrating effective targeted data augmentation techniques for medical text classification.

2. Providing methods for balancing test and training datasets.

3. Improving the overall performance of Medical Text Classifiers, particularly in handling false negatives and false positives.

These contributions aim to enhance the reliability and accuracy of NLP tools in the medical domain, ultimately benefiting healthcare professionals by providing more precise and useful text classifications.

Chapter 2

Background and Related Work

This chapter provides a brief literature review about the subject matter of this thesis. Subsequently, it describes the Clinical NLP, Transformer based models, inbalanced data phenomenon, and Active Learning framework. Finally, it summarizes the related previous work that has been done in A-PROOF project. The literature review provided in this chapter will constitute the theoretical foundation of my methodology and my experiments where I implement my methodology.

2.1 Clinical NLP and Clinical Data

The field of Natural Language Processing (NLP) has witnessed significant advancements in recent years, particularly within the realm of clinical applications. For instance, a survey by Laparra et al. (2021) highlights the interest and progress in utilizing NLP to unlock the wealth of information contained in Electronic Health Records (EHRs). EHRs are a rich source of patient data, capturing detailed narratives about patient conditions, treatments, and outcomes that are not systematically documented elsewhere.

There are many public medical datasets that have been released, focused on various standard NLP tasks. These include named entity recognition (NER) (Elhadad et al., 2015) and relation extraction (Uzuner et al., 2011), temporal information extraction (Styler IV et al., 2014), and coreference resolution (Uzuner et al., 2011). There are also datasets specifically designed for clinical tasks such as disease classification (Uzuner, 2009).

One of the primary challenges in clinical NLP is the requirement for annotated datasets. Supervised machine learning, which underpins most modern NLP methods, relies heavily on these annotated datasets. Annotating clinical text is a labor-intensive process that demands significant expertise and resources. The process involves labeling clinical narratives with medico-linguistic annotations to train NLP systems to recognize and interpret medical terminology accurately.

The survey of Laparra et al. (2021) emphasizes two critical aspects of clinical NLP: *generalizability* and *adaptability*. Generalizability refers to the ability of an NLP method to perform well on diverse test data that may differ significantly from the training data. This is crucial because clinical data can vary widely between different hospitals, regions, and patient populations. Adaptability, on the other hand, involves tailoring an initially trained model to better suit the specific characteristics of the data it will encounter at the test time. While these concepts are not mutually exclusive, they often represent

different priorities in NLP research.

Transfer learning or fine-tuning has emerged as a key strategy in enhancing both generalizability and adaptability. In transfer learning, knowledge from tasks, domains, or languages with abundant data is leveraged to improve performance in contexts where data is scarce. This approach is particularly relevant in clinical NLP, where annotated data is often limited.

As it will be be discussed in more detailed in Section 2.2 below, pre-trained transformers, such as BERT (Bidirectional Encoder Representations from Transformers) and its variants, have revolutionized NLP by achieving unprecedented performance across a range of tasks. These models are initially trained to solve general language problems using vast amounts of unlabeled data and are subsequently fine-tuned for specific downstream tasks. Despite their success, these models are not fully optimized for biomedical data. While they provide a strong foundation, there is still a need for domain adaptation techniques to tailor these models to the unique characteristics of clinical text (Laparra et al., 2021).

2.2 Transformer Models and Transfer Learning

Transformer models have revolutionized the field of natural language processing (NLP) with their ability to understand and generate human language. The transformer architecture, introduced by Vaswani et al. (2017), uses a mechanism called self-attention, which allows the model to weigh the importance of different words in a sentence when making predictions. This is a significant departure from previous models that relied heavily on sequential processing, making transformers more efficient and capable of handling long-range dependencies in text.

One of the most notable transformer models is BERT, developed by Google (Devlin et al., 2019). BERT is designed to pre-train bidirectional representations by conditioning on both left and right context in all layers, which enables it to understand the context of a word based on its surroundings. This pre-training involves two tasks: masked language modeling (predicting masked words in a sentence) and next sentence prediction (determining if one sentence follows another). BERT's ability to capture the nuances of context has set new benchmarks across various NLP tasks (Rogers et al., 2020).

RoBERTa (Robustly Optimized BERT Approach) builds upon BERT by modifying the pre-training approach. Developed by Facebook, RoBERTa uses dynamic masking instead of static masking and removes the next sentence prediction task. It also increases the amount of training data and the size of the training batches. These changes lead to significant improvements in performance over the original BERT model (Liu et al., 2019). RoBERTa has shown to be particularly effective in tasks requiring deep understanding of context and has outperformed many existing models on a variety of benchmarks (Rawat and Singh Samant, 2022).

Transfer learning, which involves fine-tuning a model pre-trained on a large dataset on a smaller task-specific dataset, is a crucial component of the success of transformer models like BERT and RoBERTa. This approach allows models to leverage knowledge from the pre-training phase, making them highly effective even with limited task-specific data. Studies have shown that fine-tuning with transformers can achieve high accuracy with relatively small annotated datasets, making it a powerful tool for tasks where data is scarce (Wankmüller, 2022).

The effectiveness of transformers in transfer learning is evident across various domains. For instance, in clinical text analysis, models like BERT and RoBERTa have been fine-tuned to identify medical conditions and annotate clinical notes with high accuracy (Yang et al., 2020).

In conclusion, transformer models such as BERT and RoBERTa have significantly advanced the capabilities of NLP through their sophisticated use of self-attention mechanisms and bidirectional context understanding. Their success is further amplified by the application of transfer learning, which allows these models to perform exceptionally well across various tasks with limited data. The continuous evolution and optimization of these models promise even greater advancements in the field of natural language processing.

2.3 Imbalanced Data

Class imbalance is defined as classification setting in which one or multiple classes (minority classes) are considerably less frequent than others (majority classes) (Henning et al., 2023). Class imbalance is a significant challenge in NLP, particularly in clinical data. This issue arises when some classes are much less frequent than others, leading to biased model performance. Transformer-based models, such as BERT and RoBERTa, although powerful, are not immune to the challenges posed by imbalanced datasets (Zhang et al., 2020).

Class imbalance often leads to models that are biased towards the majority class. This bias results in poor performance on the minority classes, which are often the most critical in clinical contexts, such as rare diseases or specific patient conditions. Studies have shown that imbalance in data can cause high false-negative rates, where the model fails to identify the minority class, leading to potentially severe consequences in clinical applications (Zhang et al., 2020).

In datasets where the negative class is much larger, models often become biased towards this majority class. This imbalance causes models to focus more on the majority class, leading to poor performance on the minority class. The learning algorithms end up primarily recognizing patterns in the majority class, increasing the chances of misclassifying the minority class, which is usually the one we are most interested in. This bias makes it harder for the model to accurately predict the minority class, resulting in higher error rates and lower overall effectiveness (Elyan et al., 2020).

In the current state of research, several typical approaches are employed to handle class imbalance in datasets. Among these, resampling techniques are the most popular and well-researched strategies. Resampling can involve either oversampling the underrepresented category or undersampling the dominant category. Oversampling strategies typically focus on generating synthetic data based on the characteristics of the underrepresented category to balance its representation in the model. For example, Khushi et al. (2021) analyzed various resampling methods and achieved promising results with certain oversampling techniques in the context of clinical data.

Oversampling with synthetic data is not the only data augmentation strategy to address imbalanced data. Henning et al. (2023) argue that increasing the amount of data for the minority class during dataset construction, such as by creating additional examples or using Active Learning to select which examples to label, can also help mitigate the class imbalance problem to some extent (Ein-Dor et al., 2020). However, this approach can be particularly labor-intensive in imbalanced settings, as it often

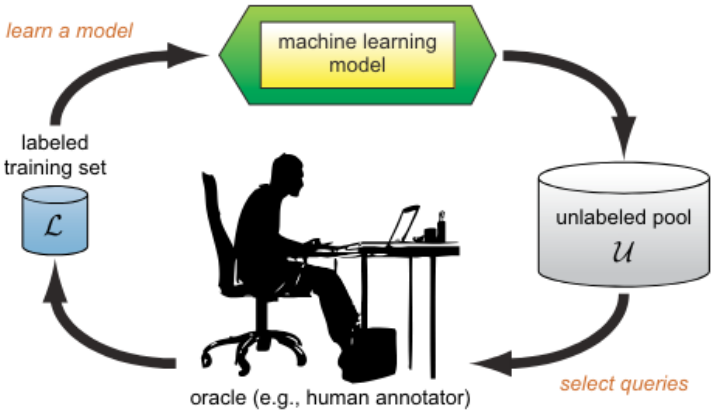


Figure 2.1: Pool Based Active Learning Setup (Settles, 2009)

requires identifying rare examples and may result in biased examples for the minority class. This will be discussed in more detailed in the Section 2.4 below.

2.4 Active Learning

Data is essential for machine learning applications and has become more valuable over time. In many cases, a large amount of unlabeled data is generated, but to use this data in supervised machine learning, it must be labeled. This often requires manual labeling, which can be complicated and might need a specialist, like in patent classification or clinical text classification. This process is time-consuming and expensive, making it impractical. Even with available experts, labeling every piece of data is often impossible due to the vast size of modern datasets. This issue is particularly significant in NLP, where both the datasets and the amount of text in each document can be enormous, leading to overwhelming annotation tasks for human experts (Schröder et al., 2022).

Active Learning (AL), is a subfield of machine learning that focuses on improving learning efficiency. The core idea is that by allowing the learning algorithm to select the data from which it learns, it can achieve better performance with fewer training instances. This property is particularly valuable because many supervised learning systems require large amounts of labeled data to perform well. While some labels, like those for spam emails or movie ratings, are easy to obtain, others can be time-consuming and costly to acquire. Active Learning addresses this challenge by querying an oracle, such as a human annotator, to label the most informative unlabeled instances. (Figure 2.1) This approach aims to maximize accuracy while minimizing the labeling effort, making it highly relevant for modern machine learning problems where data is plentiful but labels are scarce or expensive to obtain (Settles, 2009).

Active Learning reduces the amount of data that needs to be labeled by human experts. It is an iterative process between an oracle (usually a human annotator) and an active learner. Unlike passive learning, where the data is simply fed into the algorithm, the active learner decides which samples should be labeled next. The actual labeling is done by a human expert, referred to as the human in the loop. After getting new labels, the active learner trains a new model and the process starts again. The term active learner refers to the combination of a model, a query strategy, and a stopping criterion. The model is the machine learning algorithm being trained, the

query strategy determines which data points to label next, and the stopping criterion decides when to stop the labeling process (Schröder et al., 2022).

Recent advancements in NLP using AL, particularly with pre-trained models like BERT, have demonstrated remarkable performance improvements across various tasks. Ein-Dor et al. (2020) conducted a comprehensive empirical study on combining AL with BERT for text classification. Their research revealed that AL enhances BERT's performance, especially in scenarios where the data is highly imbalanced, and the annotation budget is minimal. The study examined several AL strategies: Least Confidence, which selects instances for which the model is least certain; Monte Carlo Dropout, which uses multiple inference cycles to estimate uncertainty; Perceptron Ensemble, which averages uncertainty over an ensemble of models; Expected Gradient Length, which selects instances based on the largest expected model change; Core-Set, which aims to cover the dataset space effectively; and Discriminative Active Learning, which focuses on making the labeled set representative of the entire dataset. The findings indicated that AL strategies could improve BERT's classification accuracy and F1 scores, particularly in real-world settings where initial labeled data might be biased or insufficient (Ein-Dor et al., 2020).

In this research, I will specifically use the Least Confidence strategy, which selects the most informative instances for the model from its least confident predictions. The idea behind this approach is that by validating these uncertain instances through an oracle, the model can improve its ability to generalize by learning more about unfamiliar data points. Studies by Schröder et al. (2022) and Ein-Dor et al. (2020) evaluated this basic uncertainty-based query strategy in various binary and multi-class text classification experiments, demonstrating its effectiveness for pre-trained language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019).

Another critical aspect of Active Learning is redundancy elimination with the aim of avoiding the inefficiency of labeling similar or redundant data points. When a model queries for labels, it might select multiple instances that are very similar, leading to wasted labeling effort and slower improvement in model performance. By eliminating redundant data, Active Learning ensures that each queried instance provides unique and valuable information, maximizing the efficiency of the labeling process (Jacobs et al., 2021).

Jacobs et al. (2021) explored two heuristics for redundancy elimination: *Redundancy Elimination by Training* (RET) and *Redundancy Elimination by Cosine Similarity* (RECS).

Redundancy Elimination by Training (RET) involves creating a redundancy pool (RP) from which a subset, the query pool (QP), is selected for labeling. The process begins by identifying the most uncertain instances in the RP. These instances are then incrementally retrained on, and their uncertainties are recalculated after each training iteration. This iterative retraining continues until the QP reaches the desired size, ensuring that the selected instances are diverse and informative. RET helps to ensure that the labeled set is varied and reduces the chances of repeatedly labeling similar instances, which can be computationally expensive and less informative for the model's learning process (Jacobs et al., 2021).

As for Redundancy Elimination by Cosine Similarity (RECS), unlike RET, RECS uses sentence embeddings to measure the semantic similarity between instances. By computing the cosine similarity of the sentence embeddings, RECS ensures that only those instances that are sufficiently dissimilar from each other are included in the QP.

If an instance is too similar to any already selected instance, it is excluded from the QP. This method leverages the capabilities of pre-trained models to generate embeddings that reflect the semantic content of the text, ensuring a diverse and representative set of labeled data. RECS is computationally less intensive than RET, as it does not require repeated retraining and uncertainty recalculations (Jacobs et al., 2021).

Redundancy elimination through these heuristics helps in optimizing the labeling process by ensuring that the model learns from a diverse set of instances, thereby improving its generalization ability and reducing the overall labeling cost (Jacobs et al., 2021).

In conclusion, the Active Learning framework focuses on selecting the most informative data points to label, thus improving model performance with fewer labeled instances. The AL cycle involves iteratively training the model with strategically queried, newly labeled informative data from each iteration, enhancing its learning process. Any model used in an AL setting can be trained in two ways: by updating it after each step or by re-initializing it and training it on all available annotated data (Schröder et al., 2022; Ein-Dor et al., 2020; Hu et al., 2019; Shen et al., 2018). Dodge et al. (2020) argue that the language models tend to be unstable when incrementally fine-tuned on small data sets, resulting in lower performance and higher variability. The distinction between *updating* and *re-initialization* will be quite relevant to the subject matter of this research throughout the thesis and will be discussed further in Methodology chapter (Chapter 4). However, it should still be noted that the effect of re-initializing versus updating language models during AL still remains understudied (Lemmens and Daelemans, 2023).

2.5 Previous A-PROOF Research

This section describes the previous research that has been done within the A-PROOF project. It includes the development of pre-trained MedRoBERTa.nl, the legacy fine-tuned COVID-19 model of Kim (2021), and the master theses of Badloe (2022), Galjaard (2022), Kuan (2023), and Schramm (2023). These master theses are predecessors of this research in this project and source of most of the insights incorporated here. The previous research described here can be presented successively as they all built upon each other, each contributing to the development of the project by providing new insights.

All of the master theses mentioned above built upon Jenia dataset and model that has been created through the research of (Kim, 2021). They either applied domain adaptation to Jenia model to specialize it on a specific domain, or they applied data augmentation to test the limits of the model. However, if we need to go back where everything has started, Kim's model (2021) is the first fine-tuned model based on the pre-trained transformer model *MedRoBERTa.nl* that has been created by Verkijk and Vossen (2021) and it has the largest annotated dataset that has been created so far.

2.5.1 MedRoBERTa.nl

The MedRoBERTa.nl is a Transformer based pre-trained large language model trained with Dutch medical notes (Verkijk and Vossen, 2021). The main motivation for developing MedRoBERTa.nl is the unique linguistic characteristics of Dutch clinical notes, which differs significantly from generic Dutch text data. These notes often include a

mix of general and specialized language, shorter sentence structures, and numerous medical terms not typically found in standard Dutch corpora like Wikipedia or news articles. The existing Dutch language models such as BERTje and RobBERT, which were trained on general data sources, were not adequately equipped to handle the intricacies of medical language, necessitating a domain-specific model (Verkijk and Vossen, 2021).

The pre-training of MedRoBERTa.nl included 13GB of text data sourced from Dutch Electronic Health Records (EHRs), specifically from hospital notes. This data was chosen to ensure the model could learn the specific patterns and vocabulary used in medical contexts. The RoBERTa architecture was chosen over the more typical alternative BERT due to several advantages, including the exclusion of the Next Sentence Prediction (NSP) objective, which was considered unnecessary for the tasks at hand. RoBERTa also allows for larger input sequences and utilizes a byte-level BPE tokenizer, which is beneficial for handling the varied and complex medical vocabulary (Verkijk and Vossen, 2021).

Fine-tuning the model for specific tasks, such as ICF classification, and comparing its performance to general Dutch models demonstrated that MedRoBERTa.nl excels in these tasks. This highlights the effectiveness of the domain-specific pre-training approach, as MedRoBERTa.nl showed superior performance compared to the general models (Verkijk and Vossen, 2021).

2.5.2 Jenia Model - COVID-19 Dataset

Jenia Kim's technical report, "Automated Assignment of ICF Functioning Levels to Clinical Notes in Dutch," outlines the methodologies employed in the A-PROOF project. The primary goal of this project is to develop AI models capable of identifying the functional status of patients from free-text clinical notes in Dutch. These models aim to analyze large volumes of clinical data to extract valuable insights, such as recovery patterns within specific patient populations (Kim, 2021).

ICF code	**Domain**	**Abbrev.**	**Functioning level**
b1300	Energy level	ENR	0-4
b140	Attention functions	ATT	0-4
b152	Emotional functions	STM	0-4
b440	Respiration functions	ADM	0-4
b455	Exercise tolerance functions	INS	0-5
b530	Weight maintenance functions	MBW	0-4
d450	Walking	FAC	0-5
d550	Eating	ETN	0-4
d840-d859	Work and employment	BER	0-4

Table 2.1: Overview of the ICF domains in the project

In collaboration with medical experts from the A-PROOF project, electronic health records from the Amsterdam University Medical Centers (UMC), were selected and annotated. The annotation process focused on nine *International Classification of Functioning, Disability and Health* (ICF)[1] categories shown in Table2.1. This framework

[1]https://www.who.int/standards/classifications/international-classification-of-functioning-

was established by the World Health Organization (WHO) to describe and measure health and disability in a standardized fashion.

From the 5,554 selected clinical notes, a total of approximately 286,000 sentences were annotated, with about 15,000 sentences containing at least one domain label (Kim, 2021).

The classifier was developed using a state-of-the-art NLP approach, specifically fine-tuning a pre-trained large language model. The *MedRoBERTa.nl*, was utilized as the base model. Fine-tuning was conducted using the Python library Simple Transformers[2], which facilitated the adaptation of the pre-trained *MedRoBERTa.nl* model to the specific task of ICF classification. This approach leveraged the robust language understanding capabilities of *MedRoBERTa.nl*, refined through training on extensive Dutch hospital notes, to accurately classify the functional status of patients based on their clinical records (Verkijk and Vossen, 2021). Kim's work thus contributes significantly to the field of medical NLP by providing a practical application of advanced language models in extracting meaningful health information from unstructured clinical texts (Kim, 2021).

2.5.3 Domain Adaptation Research

Domain adaptation is a technique in machine learning where a model trained on data from one domain is adapted to perform well on data from a different but related domain. This approach helps improve the model's performance in new, unseen environments by leveraging the knowledge gained from the original domain.

Oncology Data

Sharona Badloe's master thesis aimed to adapt Kim's model which was thematically partially focused on COVID-19 clinical data specifically to oncology data. For this purpose, a specialized dataset consisting of oncology notes was created, and the Jenia model was fine-tuned using this data (Badloe, 2022). Although the research provided relatively good performance on its target data, the total size of the dataset was not large enough to provide a representative test split in the end.

Primary Care Data

Ellemijn Galjaard's master thesis (2022) applied transfer learning strategy to create a classifier specialized on primary care clinical data. The data grounding the Jenia model is mainly medical notes coming from the secondary line of medical service, which is specialized clinics that patients are mostly directed to via referrals.
Galjaard's master thesis aims to adapt this model which mainly trained with secondary care data to newly curated primary care data sourced from physiotherapy, occupational therapy, and dietetics (Galjaard, 2022). The adaptation gave promising results albeit suffering from the same setbacks as Badloe's thesis (Badloe, 2022). The overall size of the dataset was not large enough to have a representative test split to evaluate the generalization performance of the model on this new domain reliably.

---

disability-and-health

[2]https://simpletransformers.ai/

2.5.4 Data Augmentation Research

Data augmentation involves expanding a dataset by creating modified versions of existing data or generating new synthetic data, which helps improve the robustness and performance of machine learning models. Kuan (2023) and Schramm (2023) both applied certain data augmentation techniques to improve the generalization capacity of the available models.

Synthetic Data via Generative LLM's

Kuan's master thesis (2023) explores the use of synthetic data generated through a generative approach to address class imbalance issues in clinical NLP. The hypothesis posits that synthetic data, when used alongside existing real data, can create a balanced training set for fine-tuning, depending on the quality of the synthetic data. By including synthetic data in the training set, Kuan aims to improve the generalizability of the classifier, thus enhancing its overall performance. The study suggests that synthetic data can mitigate the shortage of clinical data, which often arises due to privacy concerns, while also addressing class imbalance issues within datasets. This dual benefit could significantly improve the classifier's effectiveness (Kuan, 2023).

Furthermore, Kuan's work implies that synthetic data can be used to represent various medical domains during the fine-tuning phase, enabling the classifier to handle multiple tasks and thus increasing its generalizability. Presuming a positive outcome, this approach offers a promising solution to the data scarcity problem in clinical NLP. The synthetic data can complement existing real data, providing a more robust and representative training set for fine-tuning and training classifiers across different medical domains (Kuan, 2023).

Semi-Supervised Training

Cecilia Schramm's master thesis (2023) also explored innovative data augmentation techniques to enhance model performance. Schramm, specifically investigated the use of pseudo labeling, a method within semi-supervised learning, aimed at improving the model's ability to generalize (Schramm, 2023). Pseudo labeling involves the model using its own predictions as part of the training data, with the goal of uncovering new patterns that could lead to improved overall generalization and model accuracy (Xu et al., 2021).

The research of Schramm (2023) involved two main experimental setups with pseudo labeling: high-quality data and high-quantity data. In the first setup, she used high-quality pseudo-labeled data, based on targeted keywords per category. In the second setup, she employed a larger quantity of pseudo-labeled data, focusing on the volume rather than the precision of the labels. Despite the innovative approach and thorough experimentation, Schramm (2023) found that neither high-quality nor high-quantity pseudo labeling resulted in a significant improvement in the model's generalization capacity. This suggests that while pseudo labeling holds potential, its effectiveness may be limited by factors such as the inherent quality of the initial model predictions and the specific characteristics of the dataset used (Schramm, 2023).

Chapter 3

Data and Annotations

In this chapter I describe the datasets that have been used in earlier research and datasets that I curated. These datasets will be used and mentioned in the methodology (Chapter 4) and experiments (5) motivated by these methodological analyses.

3.1 Medical Notes

The medical notes data used in this study is derived from the Electronic Health Records (EHR) of patients at AUMC and VUMC. This data is rich in clinical information and encompasses a wide range of medical specializations and patient interactions. The data includes various types of clinical notes such as admission notes, discharge summaries, progress notes, and diagnostic reports.

For this particular research, I used the 'Notes' data from VUMC dated 2023. The data is presented in an unstructured format as a CSV file, with each row corresponding to one medical note and containing multiple columns with various identification details. My primary focus was on the text column, which contains the note itself, and the note ID, which assigns a unique identifier to each note. This unique ID was particularly useful when segmenting long notes into individual sentences.

Different from previous research, this thesis takes only sentences in terms of evaluation unit. All data and models are compared with respect to their sentence amounts and performance over sentences.

3.1.1 Data preprocessing

The dataset consisted of a CSV file with numerous columns containing various types of information. For this study, I focused on two columns: one containing the medical notes text and the other the note ID.

Initially, the text column underwent anonymization using SpaCy[1], where location names and person names were replaced with Named Entity Tags (i.e. PERSON, GPE). After anonymization, the text was segmented into sentences with SpaCy.

Here are the details of the data statistics for 2023 Medical Notes data:

- Total Notes: 850,000

[1] https://spacy.io/

- Total Sentences : 23 million
- Preprocessed Sentences : 7.5 million

The preprocessed sentences (7.5 million) were divided into batches of 750,000 sentences each, resulting in 7 splits of 750,000 sentences after deduplication. This division into smaller batches facilitated their separate use in different steps of the Active Learning Cycle.

For instance, in the first batch, only the first 750,000 sentences were queried for the categories ATT (attention), BER (work/employment), INS (exercise tolerance), and MBW (weight maintenance) (Figure 2.1). For the category ATT, which is more scarcely found, the first and second batches (totaling 1.5 million sentences) were queried.

This approach ensured efficient and targeted querying of the dataset during the Active Learning Cycle.

3.2 Data from Previous Research

In this section I describe the datasets that have been curated and used in previous A-PROOF research.

3.2.1 Jenia Dataset - COVID-19 Data

The dataset used in the project of Kim (2021) includes electronic health records from AUMC[2] and VUMC[3] locations. Specifically, the dataset contains approximately 4 million notes from 2017 across both locations, about 2 million notes from 2018 from the AUMC location, and another 2 million notes from the first three quarters of 2020 from both locations. Given the project's interest in COVID-19, the 2020 data is divided into two subsets: notes from patients diagnosed with COVID-19 (cov-2020) and notes from those without a COVID-19 diagnosis (non-cov-2020). The selection of a subset of these notes for annotation was critical to ensure a manageable and relevant dataset for training and evaluating the machine learning models.

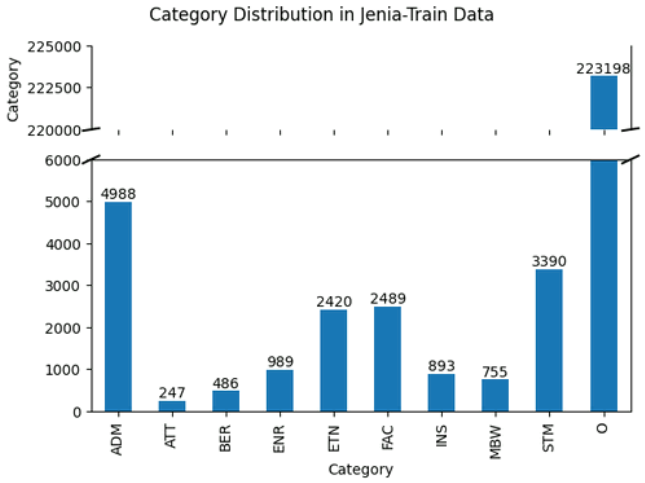


Figure 3.1: Jenia Train Split Category Distribution in Sentence Amount

[2]https://www.amsterdamumc.org/en.htm
[3]https://www.vumc.nl/

For annotation, the selected subset of data was chosen to meet several criteria to ensure quality and relevance. The annotation aimed to gather approximately 15,000 sentences with domain and level labels, ensuring a sufficient number of positive examples. The labels needed to be well-distributed across all nine domains of interest, and the sentences selected had to be diverse, encompassing various phrasings relevant to each domain. To achieve these goals, a keyword-based search was employed. This search used domain-specific keywords compiled by the core team, balancing keyword-rich and randomly selected notes to avoid bias and ensure diversity. Adjustments were made throughout the project to address imbalances in domain representation, such as varying the proportion of COVID-related notes and focusing on specific types of clinical notes to capture underrepresented domains (Kim, 2021). The category distribution in test split can be seen in Table 4.3a. Since the data is annotated for a multi-label classification task, sentences could get multiple positive values. While most sentences have one or two labels, there are a total of 65 unique label patterns, with some containing more than two labels.The pattern distribution can be seen in Table A.1.

Inter-annotator agreement (IAA)

IAA is a crucial metric for evaluating the reliability of annotated data, as it measures the consistency among different annotators. Low IAA scores indicate that there is considerable variation in how different annotators interpret and label the same data, which can undermine the quality of the dataset and subsequently affect model performance. In the given table, domains like BER, ETN, and INS exhibit particularly low IAA scores (0.42, 0.45, and 0.34, respectively), highlighting the difficulty and subjective nature of annotating these categories. This inconsistency poses a challenge for training machine learning models, as the models rely on the annotated data to learn and generalize. Therefore, when the ground truth labels are not consistently applied, the model may struggle to learn accurate representations, leading to lower performance metrics Kim (2021).

	ADM	ATT	BER	ENR	ETN	FAC	INS	MBW	STM
IAA	0.64	0.58	0.42	0.66	0.45	0.78	0.34	0.62	0.57
model	0.66	0.58	0.35	0.72	0.63	0.76	0.41	0.70	0.72

Table 3.1: F1-score: inter-annotator agreement vs. model performance (Kim, 2021)

From this point forwards, the Jenia Model and Datasets have been taken as source model to adapt and build upon in all the remaining research. When assessing the performance of any model built with this dataset, the IAA scores(Table 3.1) should be taken into account as it is a cruicial information about generalizibility capacity and potential of the models.

Jenia Test Data

The curated Jenia Dataset has been randomly split into Train/Dev/Test Splits of 80/10/10. Consequently the test split (Figure 3.2) reflects the category distribution of the whole dataset with some quite low support numbers for some underrepresented categories.

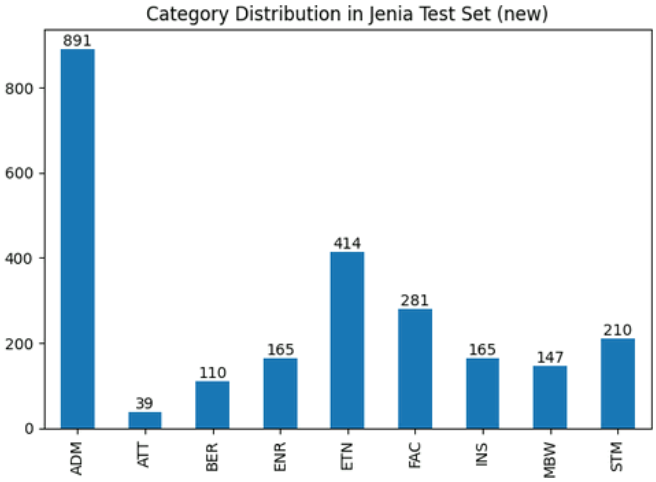


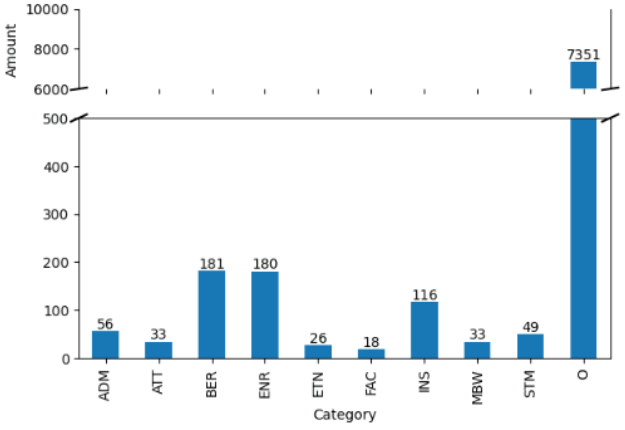
Figure 3.2: Jenia Test Split (fixed) Category Distribution in Sentence Amount

This test split is updated version with annotation fixes that have been mentioned in *Data Improvements* section 5.1.

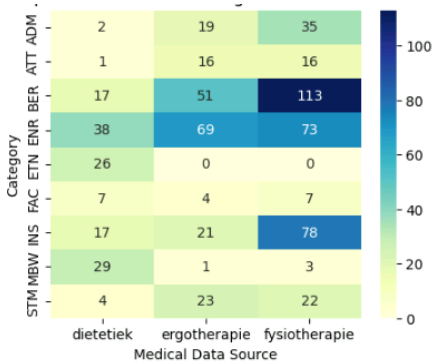
3.2.2 Primary Care Dataset - Ellemijn Data

The primary care dataset of Galjaard (2022) is sourced from three medical specializations: dietetiek (dietician), fysiotherapie (physiotherapy), and ergotherapy (occupational therapy).

The first graph (3.3a) shows the category distribution within the dataset. The negative 'O' category dominates with 7351 instances, indicating a significant portion of data not fitting into specific predefined categories. Other notable categories include BER and ENR, with 181 and 180 instances respectively, suggesting these categories are relatively well-represented compared to others like INS and STM, which have fewer instances.



(a) Category Distribution in Sentence Amount



(b) Heatmap for Medical Data Source Distribution per Category in Sentences

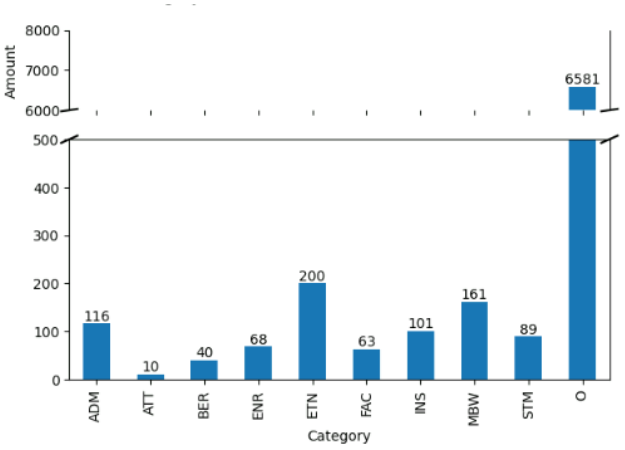
Figure 3.3: Primary Care Data Category and Source Distribution (Galjaard, 2022)

The second graph (3.3b) is a heatmap illustrating the distribution of data sources across categories. Categories are distributed thematically. Physiotherapy (fysiothera-

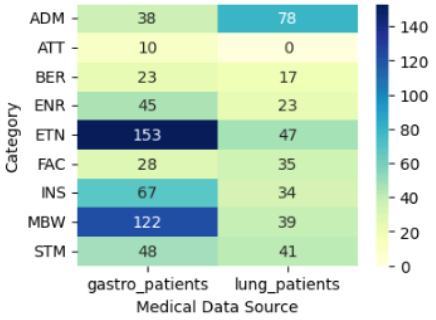
pie) contributes the most across several categories, particularly BER (working status), INS (effort) and ENR (energy), highlighting its significant presence in most physiotherapy related categories in the dataset. Occupational Therapy (ergotherapie) shows an emphasis in BER and ENR also in its category distribution as they are the most thematically related categories to the medical practice. Dietician (dietetiek), on the other hand, shows a higher number of instances in the ENR and ETN category. This distribution indicates the varying emphasis each medical specialization places on different categories within primary care documentation.

3.2.3 Oncology Dataset - Sharona Data

The Oncology Dataset curated by Badloe (2022) contains data from two primary specializations: Gastrointestinal Oncology, labeled under the batch 'gastro_patients,' and Thoracic Oncology, labeled as 'lung_patients' in the heatmap below (Figure 3.4b). The overall category distribution is consistent with the findings of Galjaard (2022), with positive categories being notably underrepresented. However, within these positive categories, there is a nuanced distribution that aligns with the thematic focus of each data source.



(a) Category Distribution in Sentence Amount



(b) Heatmap for Medical Data Source Distribution per Category in Sentences

Figure 3.4: Oncology Dataset Category and Source Distribution Badloe (2022)

Specifically, the gastrointestinal oncology data exhibits a predominant focus on categories related to eating (ETN) and weight maintenance (MBW). Conversely, the oncology data pertaining to lung cancer patients emphasizes the ADM (breathing) category. This differentiation underscores the distinct thematic concerns of each specialization, reflecting the varied clinical priorities and patient management strategies within gastrointestinal and thoracic oncology.

Chapter 4

Methodology

In this chapter, I describe the methodological aspects of my research and experiments. Initially, I briefly explain the motivations behind the data improvements implemented in this study (Section 4.1). Following that, I discuss the methodology of Fine-Tuning and its application (Section 4.2), as well as the Active Learning pipeline utilized in the experiments (Section 4.3).

4.1 Data Improvements

This section outlines the specific improvements made to the dataset to enhance the model's performance measurement as shown in Figure 4.1. It includes re-annotation of the INS(exercise) category and the correction of false-false positives, ultimately leading to a more accurate and reliable test set.

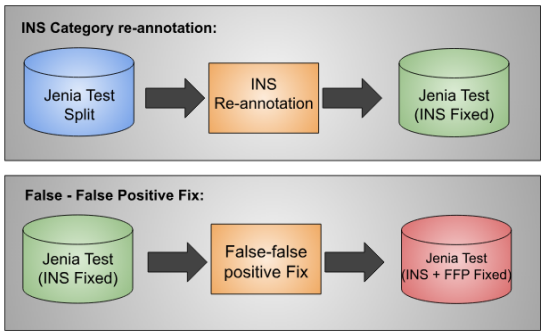


Figure 4.1: Test Data Content Improvements

4.1.1 INS Category Re-Annotation

In her master thesis, Galjaard (2022) observed certain inconsistencies in the annotation of the INS category. These inconsistencies arose either due to misunderstandings or ambiguities in the annotation guidelines, leading to some instances being incorrectly annotated as INS. Consequently, the whole INS category in both the train and the test sets were re-annotated. While this resulted in a drop in the support numbers for the category, it ultimately led to a more accurate and realistic evaluation of the classifier (Table 4.1).

Sentence	New INS Value
Heeft een nieuwe fiets, kijkt er naar uit om hierop rond te rijden, maar dit gaat helaas nog niet. *Has a new bike, looks forward to riding it, but this is not possible yet.*	0
Wel het idee dat zijn conditie achteruit is gegaan sinds de laatste embolisatie. *Has the idea that his condition has deteriorated since the last embolization.*	1

Table 4.1: INS sentences from train set, their translations, and their validated INS values

4.1.2 False-False Positives Re-Annotation

In the error analysis of her master thesis, Schramm (2023) found that some of the false positive instances did not appear to be false positives but rather true positives in the test data. In light of this finding, I decided to re-evaluate the alleged false positive instances with the medical experts in the A-Proof team. This re-evaluation led to updating the gold labels of a total of 305 instances from negative category to their respective positive category within the test data of the Jenia dataset. (Table 4.2, 5.3)

Sentence and Translation	Old Value	Updated Value
Gezien de lage belastbaarheid (zuurstofbehoefte en energiebelasting (oa zitduur)) is dit nog niet mogelijk. *Considering the low load capacity (oxygen requirement and energy load (including sitting duration)), this is not yet possible.*	O	ADM
snackbar 7 dagen/week, werkt alleen *snackbar works alone 7 days a week*	O	BER
PERSON kan zelfstandig schuiven over bed naar hoofdeinde. *PERSON can independently slide over the bed to the headboard.*	O	INS

Table 4.2: Sentences with translations, old-gold values, and updated-gold values

4.1.3 Combined Test Set

Combining separate test sets to create a balanced and more representative test set is a crucial step in ensuring the validity and generalizability of the models developed and used in the A-PROOF project. Imbalanced test sets can lead to biased performance metrics, where the model may seem to perform well on overrepresented classes but poorly on underrepresented ones. This imbalance can obscure the true capabilities of the model and potentially lead to misleading conclusions about its effectiveness.

To address this issue, I integrated the Jenia Test Split, which focuses partially on medical notes from COVID-19 patients, with the full Oncology Badloe (2022) and Primary Care datasets Galjaard (2022). The Primary Care dataset includes diverse categories such as dietician, occupational therapy, and physiotherapy, while the Oncology dataset encompasses Thoracic and Gastrointestinal Oncology patients. By combining these datasets, the resulting test set offers a more comprehensive and balanced representation of various medical conditions and specialties. This method enables the test set to assess different aspects of the healthcare data, including primary care, secondary

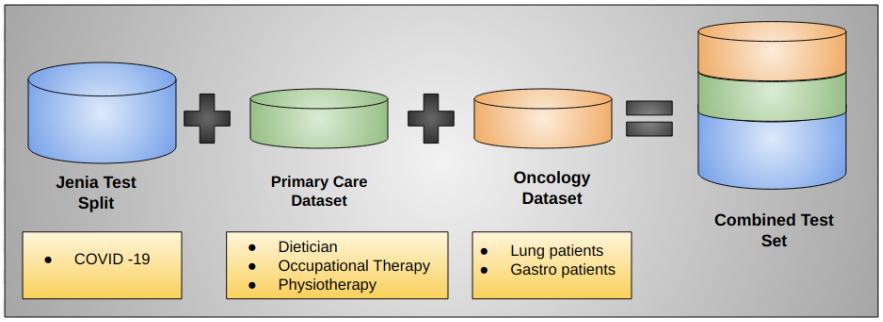


Figure 4.2: Combined Test Set composition

care, and specialized medical data such as oncology.

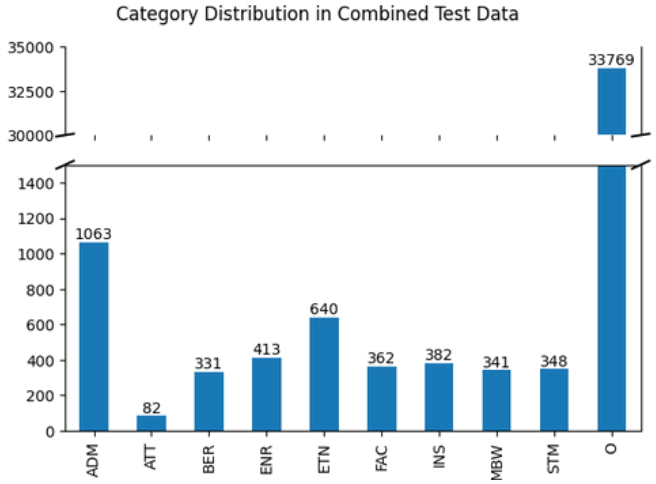
Moreover, the combination of datasets ensures that the test set includes a variety of conditions, making the evaluation process robust and reflective of real-world scenarios. This methodology helps in identifying potential weaknesses of the model when dealing with less common conditions, thereby guiding further refinement and training of the model for better overall performance. The inclusion of diverse medical data allows the model to be evaluated on multiple dimensions, encompassing various facets of patient care from primary and secondary care to specialized areas like oncology.

In conclusion, the motivation behind combining separate test sets is to create a balanced and more representative test set that accurately reflects the diverse conditions encountered in medical practice. This approach ensures that the model's performance is evaluated fairly across all classes, leading to more reliable and generalizable results. Additionally, it enhances the evaluation process by incorporating different aspects of medical data, thereby adding more dimensions for assessing the model's efficacy across various healthcare scenarios.

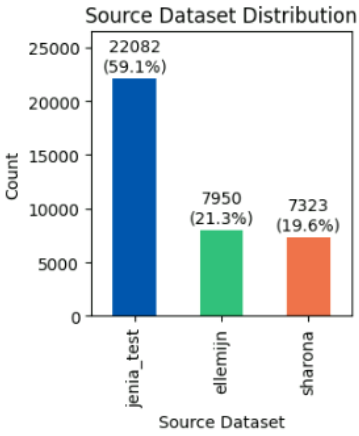
Combined Test Set Analysis

The category distribution in the combined test set varies significantly, as illustrated in Figure 4.3. The 'None' category (O) has the highest count with 33,769 instances, indicating a substantial amount of data labeled as 'None'. ADM(breathing) is the second most frequent category with 1,063 instances. Categories like ETN(eating) and ENR(eating) follow with 640 and 413 instances, respectively. ATT(attention) is the least represented category with only 82 instances. Other categories such as BER(working), FAC(walking), INS(exercise), MBW(weight maintenance), and STM(emotion) have moderate representation, ranging from 331 to 382 instances each.

The source dataset distribution is depicted in Figure 4.3b, showing the contributions from each dataset. The Jenia Test split is the largest contributor, with 22,082 instances (59.1% of the total combined set). Primary care and oncology datasets contribute 7,950 (21.3%) and 7,323 (19.6%) sentences, respectively. This distribution ensures that the combined test set benefits from a balanced mix of data from different sources in which categories are more fairly represented.



(a) Sentence Amount per Category



(b) Sentence Amount per Data Source

Figure 4.3: Combined Test Set Source and Category Distribution

A detailed breakdown of category distribution within each source dataset is provided by the heatmap in Figure 4.4. The Jenia test split predominantly contributes across all categories, particularly in ADM(breathing), ETN(eating). The primary care data Galjaard (2022) shows notable contributions in categories such as BER(working), ENR(energy), and INS(exercise). The oncology data of Badloe (2022), on the other hand, significantly contributes to ETN(eating) and MBW(weight maintenance).

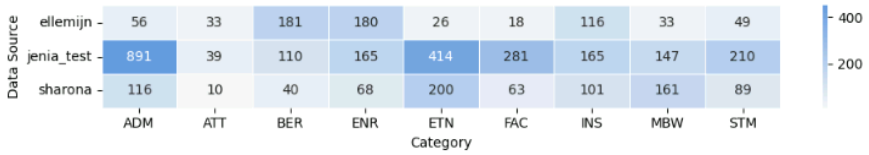


Figure 4.4: Heatmap of Category Distribution per Data Source (Sentence Amount)

In summary, the combined test set presents a well-rounded evaluation dataset with a diverse category distribution and balanced contributions from the three source datasets. This ensures that the model is tested on a variety of data types, enhancing the reliability and validity of the performance evaluation.

4.2 Trasnsfer Learning - Fine Tuning

Fine-tuning in transformer models involves adapting a pre-trained model to a specific task by training it further on a smaller, task-specific dataset. This process leverages the general language understanding developed during the initial pre-training phase and tailors it to the nuances of the new task.

4.2.1 Fine Tuning Setup

For this project, I used the MedRoBERTa.nl as the base pre-trained language model which is specifically designed for Dutch medical texts. It is pre-trained on a large corpus of Dutch hospital notes, and it provides a robust foundation for understanding and

processing medical language. MedRoBERTa.nl utilizes a byte-level Byte Pair Encoding (BPE) tokenizer, which is particularly effective for handling the unique and complex vocabulary found in medical texts (Verkijk and Vossen, 2021).

In some experiments I fine-tuned an already fine-tuned MedRoBERTa.nl model (e.g. Jenia-10), while in some of them I directly fine-tuned the base MedRoBERTa.nl. The fine-tuning process is conducted using the SimpleTransformers library[1], which simplifies the implementation of fine-tuning transformer models in Python. The key parameters for fine-tuning are empirically set as batch size *8* and learning rate *4e-5* based on hyperparameter tuning (Table 4.3).

During fine-tuning, the model was trained to classify sentences into 9 ICF categories, plus a "None" category for sentences that do not fit into any of the ICF categories.

The steps for fine-tuning involved:

- **Data Preparation:** The annotated data is formatted into the required input format for the model, with each sentence paired with its corresponding binary 10-digit category vector such as *[0,0,0,0,0,0,0,0,0,1]*.
- **Model Configuration:** The MedRoBERTa.nl pre-trained model or any of its fine tuned versions (e.g. Jenia-10) is loaded and configured with the specified batch size and learning rate.
- **Training:** The model is fine-tuned with the new prepared training data using simpletransformers python library, with the training process optimized to minimize classification errors.
- **Evaluation:** The model's performance was evaluated on a validation set to ensure it accurately classified the sentences into the correct categories.

Hyperparameter Tuning

Hyperparameters were assessed empirically by testing on validation data. Based on macro F1 scores, the optimal parameters were determined to be a batch size of 8 and a learning rate of 4e-5, as shown in Table 4.3.

Train Batch Size	**Learning Rates**		
	1e-4	**4e-5**	**1e-5**
8	0.1	**0.61**	0.59
16	X	X	0.55
24	0.1	X	0.52

Table 4.3: Hyperparameter Tuning Matrix - Macro F1 scores of Jenia-10 over dev data

In this table, 'X' indicates that the combination was not tested. From the results, it is clear that a batch size of 8 and a learning rate of 4e-5 yield the highest macro F1 score of 0.61, making this the optimal setting for the model. I also experimented with multiple epochs, yet there was no significant difference in performance. Therefore, I decided to keep all further experiments with a single epoch.

[1] https://simpletransformers.ai/

4.3 Active Learning Pipeline

The Active Learning (AL) pipeline starts with an initial model of choice, which is employed to predict over the unlabeled data. Following this, a data selection process is conducted to identify the most informative samples. These selected samples are then annotated by experts. Once annotated, the newly acquired gold-standard data is used to augment the initial dataset and subsequently to create the augmented models (Figure 4.6).

4.3.1 Classifying the Unlabeled Data

The unlabeled data is classified by taking an initial model and using it to predict over unlabeled data. I used Jenia-10 as the initial model. Jenia-10 is an updated version of the legacy model (Jenia-9), re-trained with the same training data but with 10 categories, including 'none' as a separate 10th category. This model serves as the initial model against which the performance of each iteration is compared (Section 4.4). After the first iteration, the initial Jenia train split is combined with the newly acquired AL batch, and MedRoBERTa.nl is fine-tuned with this increased training data to create the model Jenia-M1. The other intermediary model Jenia-M2 is also created with the same approach.

4.3.2 Re-training the model: Update vs. Re-initialization

As discussed in the Active Learning Section of the Background chapter above, this new gold data acquired via AL iterations can be implemented to the model in two ways (Section 2.4). They can either be created by updating the classification layer of an already fine-tuned model, or re-initializing the weights by fine-tuning the base pre-trained model with the whole data from scratch. Through 3 iterations of AL cycle I created the augmented model Jenia-M3 via re-initialization, and its alternatives Jenia-M3.1 and Jenia-M3.2 via updating as it is displayed in the figures 4.6 and 4.5.

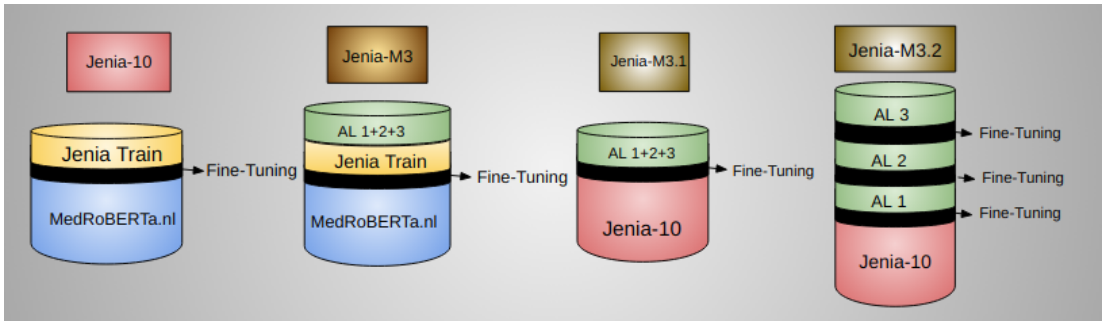


Figure 4.5: Different fine-tuning strategies behind models Jenia-10, Jenia-M3, Jenia-M3.1, and Jenia-M3.2

Re-initialization - Jenia-M3

Re-initialization is one of the fine tuning approaches where I reset the weights of the model by fine-tuning the pre-trained model with the complete augmented dataset from scratch. Specifically, I use an already fine-tuned Jenia model (Jenia-10) to retrieve

new data batches. These batches are queried and validated by experts, then they are saved as AL batches 1,2, and 3. Then I combine these newly acquired gold data batches with the original Jenia training split. Then I fine-tune the MedRoBERTa.nl pre-trained model with this new augmented training data. This process is repeated for each iteration, accumulating data progressively. The final model created using this method is referred to as *Jenia-M3*, as it results from the third iteration.

The strategy for creating iteration data batches follows this approach. The motivation for this choice is discussed in the background section above(2.4), drawing on observations by Lemmens and Daelemans (2023); Dodge et al. (2020), and Ein-Dor et al. (2020). Although the re-initialization method has not been extensively studied, the consensus suggests that continuous updating can lead to overfitting and biased results (Lemmens and Daelemans, 2023). Therefore, I have opted for re-initialization in the Active Learning cycle. Due to concerns about annotation costs, I could not experiment with continuous updating for AL iterations.

Updating - Jenia-M3.1

In this alternative approach, I fine-tuned the *Jenia-10* model instead of *MedRoBERTa.nl* pre-trained model using only the combined AL batches Batch 1,2, and 3. I call this model *Jenia-M3.1* as it stands as a fork of Jenia-M3. In this approach I only update the weights of a model that has already been fine-tuned. In other words, instead of re-initializing the weights of all layers of the pre-trained base model via fine-tuning with the complete training data, I update the weights from the previous fine-tuning process via further fine-tuning.

Continuous Updating - Jenia-M3.2

In this alternative version to updating, I fine-tuned the *Jenia-10* model in a similar fashion to Jenia-M3.1. However, on this model, instead of fine tuning the Jenia-10 with the combination of all AL batches, I fine-tuned it successively after each iteration with their respective AL batch.

4.3.3 Querying (Data Selection) Strategies

One of the essential steps in the Active Learning cycle is querying or data selection. As discussed in the section (2.4) above, the primary motivation behind Active Learning is to optimize annotation costs by selecting the most informative and unique data points from unexplored datasets, thereby maximizing performance improvement. Data selection is a critical component of the AL framework, involving the identification of data points to be fed back into the model.

In this thesis, I employed two strategies to maximize the utility of the pool of unlabeled data. First, I implemented a confidence-based selection strategy, where instances that the model classifies with the least confidence are sampled. These instances then validated by human experts (oracles) to enhance the model's understanding of less familiar data spaces. Although this sampling strategy is promising, it carries the risk of selecting repetitive patterns or similar instances, which is not ideal for the task.

To address this issue, I supplement the confidence-based selection with a cosine similarity-based clustering approach. This method aims to distill the sampled data into clusters of the most dissimilar instances, ensuring a diverse and representative set

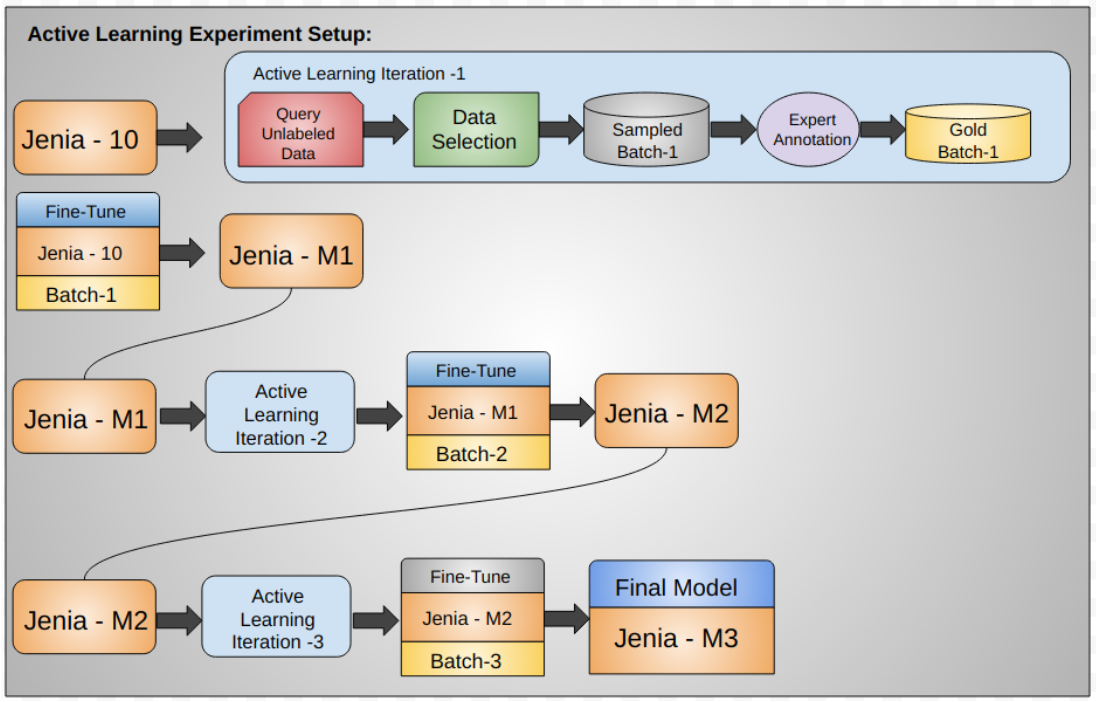


Figure 4.6: Active Learning Pipeline Schematics

of data points for annotation. This has been discussed earlier in the background section (2.4).

Confidence Based Selection

When the model predicts over the unlabeled data, it not only provides category labels but also extracts confidence scores for each category. These confidence scores are represented as a probability distribution where each value corresponds to the likelihood of a specific category. The indices of these probability values directly correspond to the indices of the categories in a predefined class representation array, such as *[1,0,0,0,0,0,0,0,0,0]* which indicates sentence is categorized as *ADM* or *[0,1,0,0,0,0,0,0,0,0]* which stands for *ATT* etc. For instance, a confidence score list such as below indicates the model's uncertainty for each category, with each probability value representing the confidence of the model that a specific category is present in the input data.

[0.14725594664970185, 0.09731612966276924, 0.020964024221496024, 0.0847652854875062, 0.052311226815738716, 0.01966332748579979, 0.01897129034805298, 0.058658743232488863, 0.034723036676526546, 0.019370829358050346]

This detailed probabilistic output allows for uncertainty-based sampling strategies, where data points with lower confidence scores can be prioritized for further annotation and training, thereby improving the model's performance on uncertain or ambiguous cases.

Having this information for each data point allowed me to calculate statistics for each category that the model predicted on the unlabeled data, as illustrated in 4.4. Using these statistics, I established a confidence range for each category in each iteration,

	Mean	Max	Min	Median
ADM	0.192508	0.214066	0.137933	0.199599
ATT	0.177317	0.201498	0.143179	0.182112
BER	0.156479	0.181988	0.144197	0.155204
ENR	0.181222	0.207282	0.1378	0.185258
ETN	0.172086	0.208895	0.136817	0.171643
FAC	0.19413	0.224223	0.135732	0.202025
INS	0.1616	0.198077	0.133001	0.157535
MBW	0.180865	0.210215	0.143126	0.183675
STM	0.191591	0.222531	0.144306	0.196488

Table 4.4: An example of confidence statistics

which guided the sampling of predictions for expert annotation.

For each category, I targeted the lower end of the confidence range, around the minimum point. In the first iteration, confidence scores were distributed more evenly, with the minimum, mean, and median points closer to each other. This allowed me to retrieve a substantial number of informative examples from the lower range for each category.

Similarity Based Selection

Active Learning aims to maximize the cost-efficiency and informativeness of data selected for expert annotation, crucial in creating high-quality, gold-standard datasets. One common approach, uncertainty-based selection, focuses on samples where the model is least confident (Section 4.3.3). However, this method can lead to repetitive patterns, reducing the diversity of the data pool. To enhance the informativeness of the dataset while maintaining cost-efficiency, it's essential to select a batch of representative, yet diverse instances. This approach ensures that each annotated instance introduces new information to the system.

To achieve this, sentences are numerically represented as word embeddings, transforming text into data points that can be compared in a multi-dimensional vector space. Pre-trained MedRoBERTa.nl represents the tokens with its embedding space. These embeddings are retrieved from the model via the simpletransformers library. By applying the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, I can identify meaningful clusters within a large sample set, particularly those with low confidence thresholds (Birant and Kut, 2007).

DBSCAN is a density-based clustering algorithm that identifies clusters based on the density of data points in a given space. Unlike other clustering algorithms, DBSCAN does not require specifying the number of clusters beforehand. Instead, it relies on two parameters: the minimum number of points (MinPts) to form a dense region and the radius (ϵ) within which points are considered neighbors (Birant and Kut, 2007).
Key features of DBSCAN include:

1. Core Points: Points that have at least MinPts within their (ϵ)-radius.

2. Border Points: Points that have fewer than MinPts within their (ϵ)-radius but are within the (ϵ)-radius of a core point.

3. Noise Points: Points that do not fit into the category of core or border points, representing outliers.

By using DBSCAN, the embedded sentences can effectively be clustered , ensuring that the selected instances for annotation are both informative and diverse. To enhance this process, I use cosine similarity to measure how similar the sentences are to each other. Cosine similarity calculates the cosine of the angle between two vectors in a multi-dimensional space, providing a measure of similarity that is independent of vector magnitude (Lahitani et al., 2016). This helps in identifying and selecting sentences that are most dissimilar, thus ensuring diversity. This combined method reduces redundancy and enhances the overall quality of the annotated dataset, leading to better model performance in subsequent training iterations.

In summary, combining word embeddings and DBSCAN clustering allows for a robust selection strategy in Active Learning, optimizing the annotation process by targeting the most informative and dissimilar instances. This approach not only improves model performance but also ensures the efficient use of expert annotators' time and resources.

4.3.4 Expert Annotation

The sampled instances presented to experts in categorized way per each batch. In order to optimize the annotation process, I asked annotators only to indicate whether the sentence actually belongs to the category that model classified inconfidently. For instance, for the first batch, I collected 156 ATT, 355 BER, 160 INS, and 160 MBW instances queried from unlabeled data (Table 5.6). I put the sentences in spreadsheets in accordance with their respective category and for each category I asked the annotator if the sentence belongs to this category or not. In this way I aimed to reduce the annotation process into a binary annotation which I presumed to be faster than usual full on multi-label annotation. In this way I acquired all validations for sentences per category and processed these validations into a new segment for training data to feed back to the model.

4.4 Models

Having described the detailed methodology, this section now briefly describes all the models used and created throughout the thesis, for the reader's convenience.

4.4.1 MedRoBERTa.nl - Pre-trained Base Model

The MedRoBERTa.nl is a Transformer based large language model pre-trained with Dutch medical notes (Verkijk and Vossen, 2021). The details about the model described in the literature review section above (Section 2.5.1). This model used as the base pre-trained model which is fine tuned for classification tasks in other domains.

4.4.2 Jenia-9 - Legacy Baseline

This model is created via fine tuning MedRoBERTa.nl with COVID-19 dataset. Since this dataset contained the biggest annotated dataset so far for ICF classification, it has been used as source domain to adapt in other research (Kim, 2021; Badloe, 2022;

Schramm, 2023). Ultimately, I will compare the performance of my best models against Jenia-9 as it is the legacy model that has not been developed further with any gold data from the same domain since it was created.

It should be noted that the Jenia-9 performance in this research appears lower than the reported performance in (Kim, 2021). I simply re-trained the MedRoBERTa.nl with the original test data again to create the model. As Kim (2021) also pointed out, this may be due to random initialization related disrepancies which may occur in transformer based models. In this thesis I take the version of Jenia-9 that went through annotation fixes to compare with Jenia-10 and other models.

In this thesis, I use the version of Jenia-9 that has undergone annotation fixes for comparison with Jenia-10 and other further models. This ensures a fair comparison between the newly developed models and the legacy model.

4.4.3 Jenia-10 - Updated Baseline

This model is created by fine tuning the MedroBERTa.nl with the Jenia Train split that has been used to create Jenia-9. This version, differently, represented the values of instances in a 10 category binary vector to correspond to 10 categories including the negative category as the 10th. The motivation behind this design change was to reduce the false positives by introducing the negative category to the model as a separate class. Therefore, it can disambiguate the negative cases better than before.

The reason that I am also taking this as another baseline is that the 10 category Jenia-10 design is used in all of other experiments in this thesis. I believe it is a better reference point than Jenia-9 to some experiments to show the differences with the same fine-tuning design.

4.4.4 Jenia-M3

This model is created via fine tuning MedRoBERTa.nl with a combination of the Jenia test split, and the Active Learning Batches 1,2,3. This model is created by re-initializing method introduced in the Methodology chapter (Section 4.3).

4.4.5 Jenia-M3.1

This model is created via fine tuning Jenia-10 with combination of Active Learning Batches 1,2,3 at once. The motivation behind this alternative experiment is to observe how the model would behave if I update the previously updated weights instead of re-fine tuning the base pre-trained model. The motivation behind this model is also discussed in the Methodology chapter above (Section 4.3).

4.4.6 Jenia-M3.2

This model, instead of fine-tuning Jenia-10 all at once with the total sum of all AL batches, fine-tunes it incrementally at each iteration with the new AL batch data. The motivation behind this model is discussed in the Methodology chapter (Section 4.3).

4.5 Evaluation

Evaluation metrics are essential tools in assessing the performance of machine learning models, particularly in medical NLP classification tasks. These metrics provide a quan-

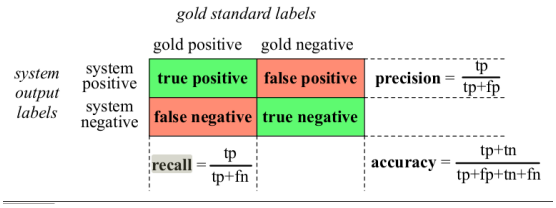


Figure 4.7: A confusion matrix for visualizing Evaluation Metrics (Jurafsky and Martin, 2009)

titative basis for comparing models, understanding their strengths and weaknesses, and guiding further improvements. They help ensure that the models are not only accurate but also robust and reliable across different scenarios. By leveraging evaluation metrics, researchers can objectively measure how well a model performs in various aspects, such as precision, recall, and F1-score, which are crucial for making informed decisions about model deployment in real-world medical applications.

Precision is the ratio of correctly predicted positive observations to the total predicted positives. It is a measure of the accuracy of the positive predictions. Mathematically, it is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Where TP represents True Positives (correctly predicted positive cases) and FP represents False Positives (incorrectly predicted positive cases) (Jurafsky and Martin, 2009).

Recall, also known as sensitivity or true positive rate, is the ratio of correctly predicted positive observations to all the observations in the actual class. It measures how well the model can identify all relevant instances. Mathematically, it is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where FN represents False Negatives (actual positive cases that were incorrectly predicted as negative) (Jurafsky and Martin, 2009).

The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is particularly useful when you need to take both false positives and false negatives into account. Mathematically, it is defined as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Macro F1 Score is the arithmetic mean of the F1 Scores of all classes. It treats all classes equally by calculating the F1 Score for each class and then averaging them. This metric does not take into account the class imbalance, treating each class as equally important (Jurafsky and Martin, 2009).

$$\text{Macro F1} = \frac{1}{N} \sum_{i=1}^{N} \text{F1 Score}_i$$

Where N is the number of classes.

Micro F1 Score aggregates the contributions of all classes to compute the average F1 Score. It considers the total true positives, false negatives, and false positives across all classes to calculate precision and recall, and then derives the F1 Score from these aggregated values. This metric is more sensitive to class imbalance, reflecting the performance on the dataset as a whole (Jurafsky and Martin, 2009).

$$\text{Micro F1} = \frac{2 \times \text{Micro Precision} \times \text{Micro Recall}}{\text{Micro Precision} + \text{Micro Recall}}$$

Where Micro Precision and Micro Recall are calculated as:

$$\text{Micro Precision} = \frac{\sum TP_i}{\sum TP_i + \sum FP_i}$$

$$\text{Micro Recall} = \frac{\sum TP_i}{\sum TP_i + \sum FN_i}$$

These metrics are essential for evaluating the performance of classification models, helping to understand their strengths and weaknesses in terms of both detecting relevant instances and avoiding false alarms. In my research I used the metric calculation tools provided by sciki-learn python library.[2]

[2]https://scikit-learn.org/stable/api/sklearn.metrics.html#module-sklearn.metrics

Chapter 5

Experiments and Results

This chapter provides an overview of the experiments conducted, along with a detailed analysis of the results and relevant comparisons. As discussed in the introduction above (Section 1), having misannotations in the train and test data makes it difficult to assess the genuine generalization capacity and performance of the model. This section demonstrates the results of implemented data improvements, such as the re-annotation of the INS category and the False-false Positive (FFP) instances. Additionally, the chapter evaluates the performance of the model that has been redesigned to incorporate 10 categories.

Following that the chapter explores the implementation of the Active Learning process to further refine and optimize the previously available models' performance. The AL experiments include one model created by re-initialization (Jenia-M3) (Section 4.3.2), two models created by updating (Jenia-M3.1, Jenia-M3.2) (Section 4.3.2, 4.3.2). I compare these models created as a result of AL cycle against both old test data and new combined test data(Section 4.1.3).

Finally, I explored the potential of the models through minor experiments. These included pseudo-labeling, where the model's predictions were used to label unlabeled data, and fine-tuning with different data compositions, including the addition of negative instances. These experiments aimed to assess and enhance the models' robustness and accuracy by testing various data augmentation techniques and training strategies.

5.1 Data Improvements

This section provides an overview of the key data improvements made during the project. It begins with the re-annotation of the INS (exercise tolerance) category, addressing previously identified inconsistencies and leading to a more accurate evaluation of the classifier. Then it describes the FFP issue and how it has been addressed.

5.1.1 INS Data

It can be seen in Table 4.1 that the value of some examples has changed from 1 to 0, while others have remained as 1. As a result, the total support number for the category has dropped from 1967 to 893 in the training data and from 287 to 148 in the test data, as shown in Table 5.1

When comparing the INS performance of the Jenia model re-trained with the fixed INS category, an increase in recall is observed (Table 5.2). Despite this improvement in

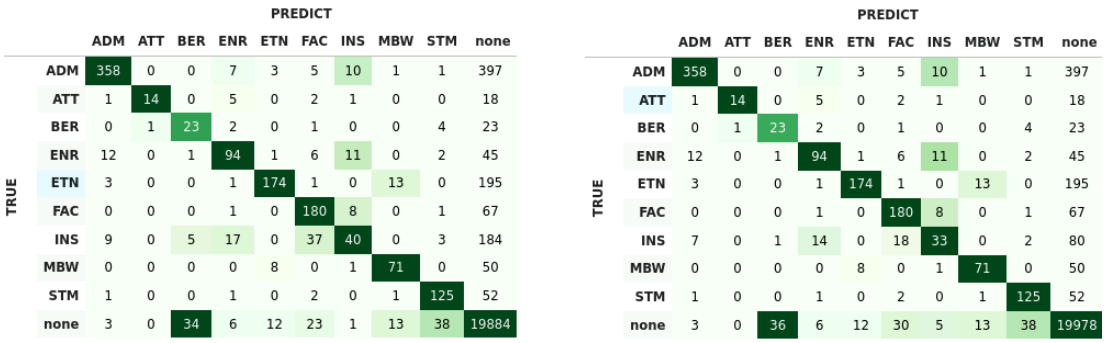
recall, there is a slight drop in precision. This decrease in precision may be attributed to the significant drop in support numbers for the INS category in the training data.

	Old #	**New #**
Train	1967	893
Test	287	148

Table 5.1: Old and New INS instance counts in Jenia Dataset after re-validation of the category

Category	**Old INS**				**New INS**			
	Precision	**Recall**	**F1**	**Support**	**Precision**	**Recall**	**F1**	**Support**
ADM	0.99	0.46	0.63	775	0.99	0.46	0.63	775
...	...	...	...	...	...	...	...	...
INS	**0.93**	**0.14**	**0.24**	**287**	**0.77**	**0.22**	**0.35**	**148**
...	...	...	...	...	...	...	...	...
STM	0.76	0.69	0.72	181	0.76	0.69	0.72	181

Table 5.2: Comparison of metrics for model trained with fixed INS data on old and new INS test data, Model: Jenia-9(INS fixed)



(a) Old INS Test Data

(b) Updated INS Test Data

Figure 5.1: Confusion Matrices of fixed INS model on old and updated INS Test data

It is important to note that making a clear comparison between the old and new models is challenging. The old model, which was trained with mislabeled INS values, cannot be considered reliable. Naturally, the model re-trained with fixed INS values performs better with a test set that has been corrected in the same manner. Assessing the improvement is difficult, as comparing a wrongly trained model against either correct or incorrect test data, or a correctly trained model against either correct or incorrect test data, would not provide clear insights.

My objective was to re-establish the external consistency of the INS category with expert validation, rather than relying on its unrealistic internal consistency. This approach ensures that the evaluation aligns more accurately with practical expectations and knowledge. This model with fixed INS will be taken as reference for the next experiment of false-false positive fixes.

5.1.2 False False Positives

When I adjusted the gold labels of false-false positives in the test data, several changes in the evaluation metrics were observed, as shown in Table 5.4.

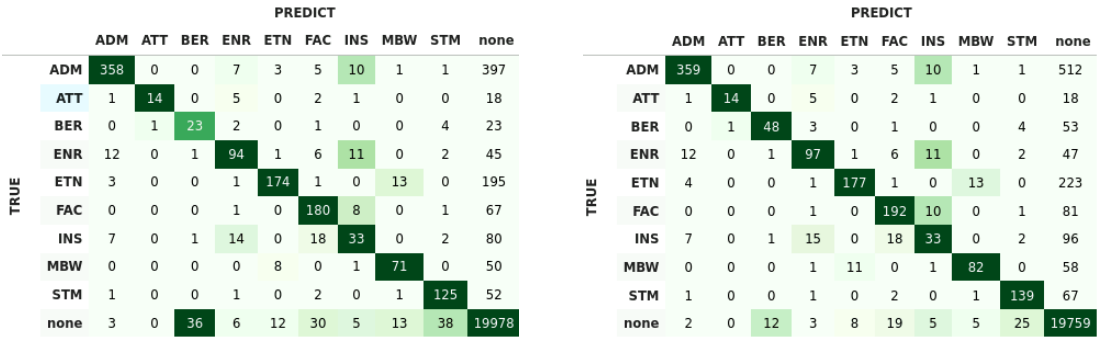
Category	Old	FFP	New
ADM	775	116	891
ATT	39	0	39
BER	54	56	110
ENR	160	5	165
ETN	382	32	414
FAC	253	28	281
INS	148	17	165
MBW	125	22	147
STM	181	29	210
O	20120	-	19897
Total	22077	305	22136

Table 5.3: Amount of Fixed FFP instances in Jenia Test Data

Overall, the adjustments to the gold labels of false-false positives in the test data led to notable improvements in precision for several categories, particularly BER and MBW. While some categories experienced slight decreases in recall, the overall impact on F1 scores was mixed, with some categories showing improvements and others remaining stable. As it can also be observed in confusion matrices (Figure 5.2), the re-annotation effort primarily enhanced the model's ability to correctly identify true positives without substantially compromising its recall performance.

Category	Old Test (FFP Unfixed)				FFP Fixed				
	Precision	Recall	F1	Support	Precision	Recall	F1	Diff	Support
ADM	0.99	0.46	0.63	775	0.99	0.40	0.57	116	891
ATT	1	0.36	0.53	39	1	0.36	0.53	0	39
BER	0.38	0.43	0.40	54	**0.80**	**0.44**	**0.56**	56	110
ENR	0.92	0.59	0.72	160	**0.95**	0.59	**0.73**	5	165
ETN	0.93	0.46	0.61	382	**0.94**	0.43	0.59	32	414
FAC	0.85	0.71	0.77	253	**0.90**	0.68	**0.78**	28	281
INS	0.77	0.22	0.35	148	0.77	0.20	0.32	17	165
MBW	0.80	0.57	0.66	125	**0.92**	0.56	**0.69**	22	147
STM	0.76	0.69	0.72	181	**0.84**	0.66	**0.74**	29	210

Table 5.4: Comparison of metrics for test data with fixed and unfixed false-false positive values, Model:Jenia-9 (INS Fixed)



TRUE \ PREDICT	ADM	ATT	BER	ENR	ETN	FAC	INS	MBW	STM	none
ADM	358	0	0	7	3	5	10	1	1	397
ATT	1	14	0	5	0	2	1	0	0	18
BER	0	1	23	2	0	1	0	0	4	23
ENR	12	0	1	94	1	6	11	0	2	45
ETN	3	0	0	1	174	1	0	13	0	195
FAC	0	0	0	1	0	180	8	0	1	67
INS	7	0	1	14	0	18	33	0	2	80
MBW	0	0	0	0	8	0	1	71	0	50
STM	1	0	0	1	0	2	0	1	125	52
none	3	0	36	6	12	30	5	13	38	19978

(a) Test data unfixed FFP

TRUE \ PREDICT	ADM	ATT	BER	ENR	ETN	FAC	INS	MBW	STM	none
ADM	359	0	0	7	3	5	10	1	1	512
ATT	1	14	0	5	0	2	1	0	0	18
BER	0	1	48	3	0	1	0	0	4	53
ENR	12	0	1	97	1	6	11	0	2	47
ETN	4	0	0	1	177	1	0	13	0	223
FAC	0	0	0	1	0	192	10	0	1	81
INS	7	0	1	15	0	18	33	0	2	96
MBW	0	0	0	1	11	0	1	82	0	58
STM	1	0	0	1	0	2	0	1	139	67
none	2	0	12	3	8	19	5	5	25	19759

(b) Test data fixed FFP

Figure 5.2: Confusion Matrices of Test Data with unfixed and fixed false-false positives, Model:Jenia-9

The drop in false positives can particularly be observed in BER, FAC, and MBW categories in confusion matrices (Figure 5.2.

5.2 Introducing 10th Category - Jenia-10

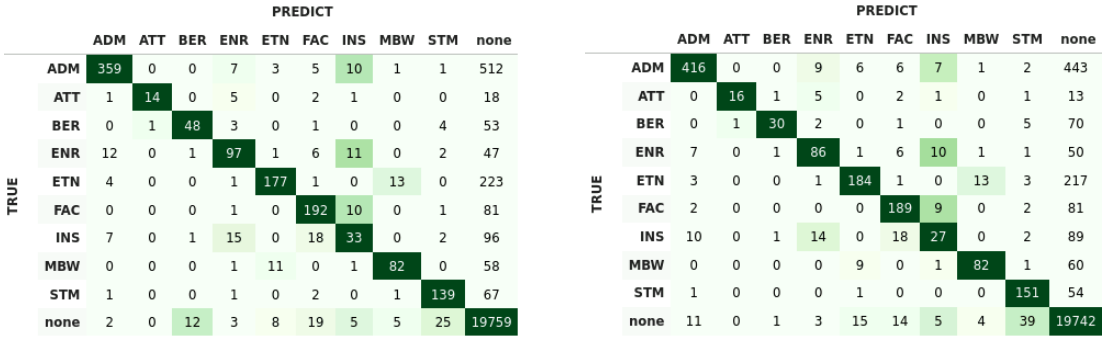
In all previous experiments, including the legacy model by Kim (2021), the model was trained with only 9 ICF categories. To address the false positive behavior observed in the model (Figure 5.3), I decided to re-train the model with an additional, tenth category, representing negative instances. By incorporating this negative category into the training process, I anticipated that the model would learn to recognize and classify the 'none' category more effectively. This adjustment was aimed at reducing the false positive rate and improving the overall performance of the model.

The Jenia-9 model is the baseline model with fixed INS category, and the Jenia-10 model is the model trained with the same data but with 10 category including 'none'. The test data is the one with fixed false-false positives and fixed INS annotations.

In the classification report (Table 5.5) introducing the negative category into the model training results in a slight improvement in recall. This enhancement is also evident in the confusion matrices in Figure 5.3. Specifically, for the categories ADM, ATT, ETN, and STM, there is an observed improvement in recall, leading to a reduction in false negatives. However, for the categories BER, ENR, and MBW, recall did not improve, but there was an improvement in precision.

Category	Jenia-9				Jenia-10			
	Precision	**Recall**	**F1**	**Support**	**Precision**	**Recall**	**F1**	**Support**
ADM	**0.99**	0.40	0.57	891	0.97	**0.47**	**0.63**	891
ATT	1	0.36	0.53	39	1	**0.41**	**0.58**	39
BER	0.80	**0.44**	**0.56**	110	**0.97**	0.27	0.43	110
ENR	0.95	0.59	**0.73**	165	**0.97**	0.52	0.68	165
ETN	**0.94**	0.43	0.59	414	0.92	**0.44**	**0.60**	414
FAC	0.90	0.68	0.78	281	**0.92**	0.67	0.78	281
INS	0.77	**0.20**	0.32	165	0.77	0.16	0.27	165
MBW	0.92	0.56	0.69	147	**0.93**	0.56	**0.70**	147
STM	0.84	0.66	0.74	210	0.78	**0.72**	**0.75**	210

Table 5.5: Comparison of metrics for Jenia-9 baseline model and Jenia-10 model, Test data: Jenia Test split (FFP,INS Fixed)



(a) Jenia-9 (b) Jenia-10

Figure 5.3: 9 category and 10 category models on fixed Jenia Test Data

5.3 Active Learning

This section describes the process of query strategies adopted throughout the Active Learning (AL) cycle and model development through each iteration of the cycle. The section begins with an overview of data selection techniques. In this part, I detail the strategies used to choose data samples that enhance model performance. Specifically, I discuss Uncertainty Based Sampling, where samples that the model is least confident about are selected for labeling and further training. Additionally, we cover Cosine Similarity Based Clustering, an approach that involves clustering data based on cosine similarity to ensure that diverse and representative samples are chosen for training.

Following data selection, the section moves on to Hyperparameter Tuning. Here, I describe the methods employed to optimize the model's hyperparameters, which are crucial for improving both performance and efficiency. The fine-tuning of these parameters ensures that the model operates at its best possible capacity.

Finally, the section concludes with a detailed look at the Active Learning Cycle itself. This subsection outlines the iterative process of Active Learning, which includes steps such as data selection, model training, evaluation, and refinement. Each iteration of this cycle is designed to progressively enhance the model's capabilities, ensuring continuous improvement and adaptation to new data.

5.3.1 Data Selection

The Table 5.6 presents the results from three batches of data retrieved through Active Learning iterations. Each batch includes various categories of data, the confidence range of predictions, the number of instances, the correctness of predictions after validation by human experts, and whether cosine similarity clustering was applied.

In Batch-1, the focus was on categories ATT, BER, INS, and MBW, with no clustering applied. Batch-2 expanded to include additional categories and applied clustering to some. Batch-3 targeted all categories proportionally to their representation in the training data, with no clustering used.

Overall, clustering in Batch-2 did not significantly improve correctness, and there was considerable variability in accuracy across categories and batches. The decrease in correctness from Batch-1 to Batch-3 highlights the need for refined data selection strategies and potential improvements in the model to enhance performance. Additionally, models may have overfitted to the data, reducing the proportion of correct predictions at lower confidence levels as iterations progressed.

Uncertainty Based Sampling

On each iteration of the AL cycle, the model made predictions on a different batch of unlabeled data consisting of 750,000 sentences. The goal was to identify informative low-confidence instances to validate and feed back into the system. Based on the confidence statistics from the total predictions, as shown in Table 5.7, the lower end of the confidence range was targeted.

In Batch-1, the Jenia-10 model was used to predict over the unlabeled data. The initial focus was on the most underrepresented categories: ATT, BER, INS, and MBW. A total of 831 sentences were selected for these categories, with approximately half being accurate predictions after expert validation.

For Batch-2, I used Jenia-M1, an augmented version of Jenia-10 with Batch-1 data. The category range was expanded to include all categories proportionately, meaning well-represented categories were sampled less, and underrepresented categories more. This batch aimed for a wider range of confidence space to increase the number of positive predictions and to experiment with informative clustering. However, after validation, the accuracy of predictions dropped, with only 22% of the sampled instances being correct.

In Batch-3, a similar confidence range was used, targeting values between the mean and minimum. The intermediary model Jenia-M2, an augmented version of Jenia-M1 with Batch-2 data, was used. The sample size per category varied according to category representation in the initial model dataset. Clustering was not applied in this batch due to its lack of promising results in Batch-2. Despite increasing the number of sampled instances, the proportion of positive validated instances decreased (Table 5.6).

Cosine Similarity Based Clustering

In the process of applying DBSCAN clustering to the queried data, I encountered significant challenges due to the nature of pairwise cosine similarity distances observed within the data. To illustrate this, I generated a histogram of these pairwise distances, which revealed a predominance of very low distance values (Figure A.1). This distribution indicates a high degree of similarity between many data points, complicating the clustering process based on mean cosine similarity.

Batch	Category	Confidence	Instances	Correct/Total	Correct	Clustered
Batch-1	ATT	0.16-0.12	156	105/156	0.67	No
	BER	0.14-0.12	355	110/355	0.3	No
	INS	0.14-0.12	160	110/160	0.68	No
	MBW	0.14-0.12	160	106/160	0.66	No
Batch-1 Total			831	431/831	0.51	
Batch-2	ADM	0.17-0.10	83	51/83	0.61	Yes
	ATT	0.17-0.11	573	47/573	0.08	No
	BER	0.17-0.12	180	5/180	0.02	Yes
	ENR	0.16-0.12	70	62/70	0.88	Yes
	ETN	0.16-0.12	119	64/119	0.53	Yes
	FAC	0.16-0.12	62	25/62	0.40	Yes
	INS	0.14-0.12	202	39/202	0.19	Yes
	MBW	0.14-0.12	128	23/128	0.17	Yes
	STM	0.16-0.12	37	15/37	0.40	Yes
Batch-2 Total			1454	331/1454	0.22	
Batch-3	ADM	0.16-0.12	23	14/23	0.63	No
	ATT	0.16-0.12	500	5/500	0.01	No
	BER	0.14-0.12	500	6/500	0.01	No
	ENR	0.16-0.12	89	17/89	0.19	No
	ETN	0.16-0.12	112	42/112	0.375	No
	FAC	0.16-0.12	121	61/121	0.5	No
	INS	0.14-0.12	392	48/392	0.12	No
	MBW	0.14-0.12	277	10/277	0.03	No
	STM	0.16-0.12	114	65/114	0.57	No
Batch-3 Total			2128	268/2128	0.12	

Table 5.6: Statistics of queried data sent for expert annotation

My preprocessing steps involved several key actions aimed at refining the dataset. Initially, I removed all words shorter than four characters. This step was crucial to eliminate potentially noisy and less informative words that might skew the embedding space. Next, I removed duplicate entries to ensure that each sentence is unique, thus maintaining the integrity of the clustering process. Following these steps, I computed the mean word embeddings for each sentence to represent them in a high-dimensional space suitable for clustering.

Despite these efforts, the application of the DBSCAN algorithm faced difficulties. DBSCAN is designed to identify clusters based on density, requiring a careful balance of its parameters, epsilon (ϵ) and the minimum number of points (minPts). However, the overwhelmingly low pairwise distances meant that setting an appropriate ϵ value was challenging. A too-small ϵ would result in many points being labeled as noise, while a too-large ϵ would lead to a single large cluster, failing to differentiate meaningful subgroups within the data.

This high density of points within a narrow distance range likely stems from the high similarity between many text entries, potentially due to the nature of the medical text data used. Such data often contains repeated terminology and phrases, contributing to the observed distribution. Consequently, the DBSCAN algorithm struggled to distinguish distinct clusters, underscoring the need for alternative strategies or additional

Batch	Category	Max	Min	Mean	Median	Sampled Range
Batch-1	ATT	0.2205	0.1169	0.1847	0.1843	0.16-0.12
	BER	0.2163	0.1028	0.1799	0.1799	0.14-0.12
	INS	0.2061	0.1112	0.1682	0.1675	0.14-0.12
	MBW	0.2166	0.1113	0.1899	0.1986	0.14-0.12
Batch-2	ADM	0.2174	0.1210	0.1886	0.1939	0.17-0.10
	ATT	0.2219	0.1175	0.1818	0.1829	0.17-0.11
	BER	0.2184	0.1133	0.1803	0.1795	0.17-0.12
	ENR	0.2153	0.0993	0.1917	0.2032	0.16-0.12
	ETN	0.2097	0.0968	0.1846	0.1906	0.16-0.12
	FAC	0.2210	0.1285	0.1930	0.2011	0.16-0.12
	INS	0.2072	0.0890	0.1694	0.1669	0.14-0.12
	MBW	0.2212	0.1158	0.1871	0.1908	0.14-0.12
	STM	0.2228	0.1305	0.1989	0.2079	0.16-0.12
Batch-3	ADM	0.2174	0.1210	0.1886	0.1939	0.17-0.10
	ATT	0.2219	0.1175	0.1818	0.1829	0.17-0.11
	BER	0.2184	0.1133	0.1803	0.1795	0.17-0.12
	ENR	0.2153	0.0993	0.1917	0.2032	0.16-0.12
	ETN	0.2097	0.0968	0.1846	0.1906	0.16-0.12
	FAC	0.2210	0.1285	0.1930	0.2011	0.16-0.12
	INS	0.2072	0.0890	0.1694	0.1669	0.14-0.12
	MBW	0.2212	0.1158	0.1871	0.1908	0.14-0.12
	STM	0.2228	0.1305	0.1989	0.2079	0.16-0.12

Table 5.7: Statistics of queried data sent for expert annotation, models used to retrieve confidence scores are Batch-1:Jenia-10, Batch-2:Jenia-M1, Batch-3:Jenia-M2)

preprocessing steps to better separate the data in the embedding space.

The lower confidence samples that I was interested in were relatively small, ranging between 100 to 500 instances. When I attempted clustering these samples, I could not achieve any meaningful grouping. I only managed to obtain a tangible number of clusters by empirically adjusting the ϵ and minPts parameters when working with relatively large samples. For example, with 2000 instances, I could generate 50-80 clusters. However, these larger sample sizes represented well-represented categories, which were not my main focus. Consequently, I could only use clustering to extract the most informative pieces from these large samples, rather than from the smaller, less confident samples that were of primary interest to me.

5.3.2 Active Learning Cycle

The table presents the performance metrics for different categories across the iterations of Active Learning cycle. Each iteration (M1, M2, M3) represents a model trained with progressively larger datasets, starting from the base model (M0), and adding Batch-1, Batch-2, and Batch-3 sequentially. Models tested agains development data, in this process. Here is a detailed interpretation:

ADM (Breathing): Precision increased until Batch-2, Recall fluctuated.

Table 5.8: Performance Metrics for Each Category on each Active Learning iteration - Model-M0: Jenia-10, Model M1,M2,M3 models per iteration with their increased instances, testing on dev set

Category	Added	Precision	Recall	F1-Score
ADM-M0	0	0.68	**0.70**	**0.69**
ADM-M1	3	0.69	0.66	0.68
ADM-M2	51	**0.73**	0.64	0.68
ADM-M3	120	0.68	0.69	0.68
ADM-Total	174			
ATT-M0	0	**0.78**	0.32	0.45
ATT-M1	105	0.75	**0.55**	**0.63**
ATT-M2	47	0.73	0.55	0.63
ATT-M3	8	0.71	0.55	0.62
ATT-Total	160			
BER-M0	0	**0.54**	0.48	0.51
BER-M1	110	0.44	0.59	0.50
BER-M2	5	0.50	**0.62**	**0.55**
BER-M3	15	0.44	0.62	0.51
BER-Total	140			
ENR-M0	0	**0.75**	0.67	0.71
ENR-M1	5	0.75	**0.70**	**0.73**
ENR-M2	62	0.74	0.70	0.72
ENR-M3	62	0.74	0.70	0.72
ENR-Total	129			
ETN-M0	0	0.50	0.62	0.55
ETN-M1	5	**0.54**	**0.63**	0.58
ETN-M2	114	0.53	0.61	0.57
ETN-M3	115	0.52	0.61	0.57
ETN-Total	234			
FAC-M0	0	0.44	0.67	0.54
FAC-M1	1	0.49	**0.79**	**0.60**
FAC-M2	25	**0.60**	0.76	0.60
FAC-M3	140	0.47	0.76	0.58
ETN-Total	166			
INS-M0	0	0.61	0.09	0.15
INS-M1	110	0.45	0.15	0.22
INS-M2	39	**0.62**	0.16	0.25
INS-M3	96	0.61	**0.18**	**0.28**
INS-Total	245			
MBW-M0	0	**0.76**	0.74	0.75
MBW-M1	106	0.73	**0.77**	0.75
MBW-M2	23	0.72	0.75	0.73
MBW-M3	40	0.71	0.75	0.73
MBW-Total	169			
STM-M0	0	0.52	0.69	0.59
STM-M1	0	0.55	0.67	0.60
STM-M2	15	**0.56**	0.73	0.63
STM-M3	102	0.54	**0.76**	**0.63**
STM-Total	117			

ATT (Attention): Precision only decreased, recall spiked at Batch-1 then stabilized.

BER (Work): Precision only decreased, recall increased gradually util Batch-2.

ENR (Energy): All metrics remained almost steady.

ETN (Eating): With Batch-1 improvement observed in both Precision and Recall together, then both decreased.

FAC (Walking): Recall peaked at Batch-1, Precision at Batch-2, then both dropped at Batch-3.

INS (Exercise Tolerance): recall steadily increased, Precision fluctuated.

MBW (Weight Maintenance): Precision gradually decreased, recall slightly improved at Batch-2 with increased support number, then decreased again.

STM (Emotion): Recall increased significantly at Batch-3, Precision at Batch-2.

Overall, the Active Learning cycle demonstrates that adding more data with re-initialization technique did not show a notable improvement throughout the AL iterations (4.3.2). The results have small fluctuations and small improvements in recall in some categories.

5.4 Active Learning Batches as the New Data

In this section I will describe the datasets that I have created via AL.

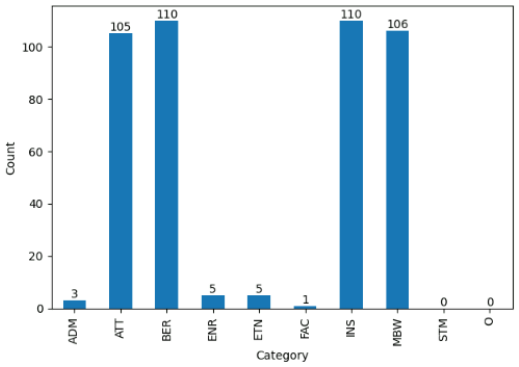
This section shows the positive instances retrieved per each AL iteration. The number of instances retrived per category are higher than the ones in Table 5.6, because I asked annotators also to mark other categories if present in a given sentence. Therefore, there are notable amount of unintended positive cases within the batches. The intended instance here means that

5.4.1 Batches per Iteration

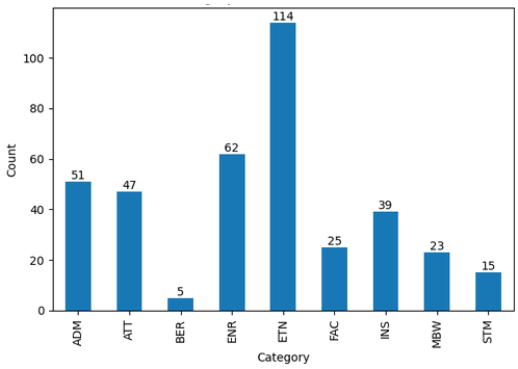
As shown in Table 5.6, the first batch yielded 431 positive instances from the targeted categories. Some of these instances had multiple labels per sentence, consequently including additional positive instances.

Category	Intended	Unintended	Total
ADM	14	106	120
ATT	5	3	8
BER	6	9	15
ENR	17	45	62
ETN	42	73	115
FAC	61	79	140
INS	48	48	96
MBW	10	30	40
STM	65	37	102

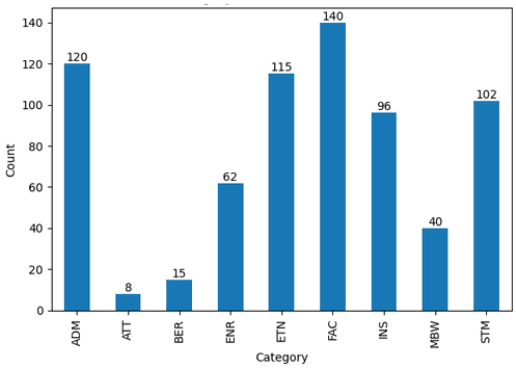
Table 5.9: Intended, Unintended, and Total Instances per Category in Batch-3 (Sentence)



(a) Batch-1 Total: 440 Sentences



(b) Batch-2 Total: 381 Sentences



(c) Batch-3 Total: 506 Sentences

Figure 5.4: Category distribution of retrieved positive instances from 3 AL iteration Batches

The second batch contains 381 instances, including 50 unintended ETN instances, which are added to the 331 intended ones.

The third batch included a significant number of unintended instances. Here the term 'intended' stands for positive instances coming from their category specific annotation sets. For example, from queried low confidence ADM instances I got only 14, yet instances in other categories somehow included 106 ADM instances, either via multi-label instances or misclassified instances. Initially, the intended amount was 268, but the total ended up being 506 sentences. This is detailed in Table 5.9, which shows the distribution of intended, unintended, and total instances across various categories.

5.4.2 Final Augmented Train Data

This section demonstrates the final state of the category distribution in training data after targeted augmentation techniques applied throughout the thesis and compares it with the old data category distribution. The left bars in the plot on each category represents the old train data with its amount of sentences and their percentage within their dataset. The right bars represent the new dataset in the same way.

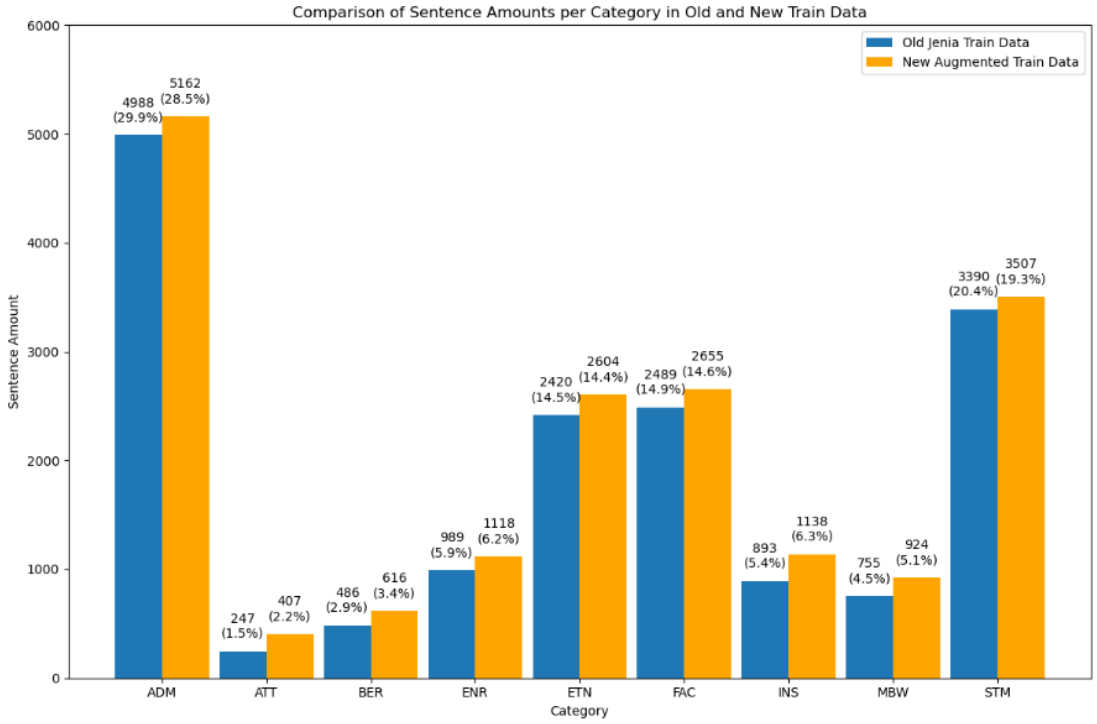


Figure 5.5: Comparison of Sentence Amounts per Category in Old and New Train Data

It can be seen from the graph 5.5 that the new data brings about a slightly more balanced category distribution. While the portion of previously underrepresented categories increased, the share of dominant categories within the dataset slightly reduced. For example, an increase can be observed ranging from .3% to .9% in categories ATT, BER, ENR, INS, and MBW. This number would be higher if I did not label and query sufficiently represented categories in the dataset, yet I included them in the experiments too.

5.5 Testing Models on Combined and Jenia Test Data

In this section, I evaluate the created models (M3, M3.1, M3.2) against both the legacy Jenia test data and the newly combined test data. These evaluations also include Jenia-10 baseline comparisons. This section aims to highlight the results related to the main objectives of this thesis, particularly the concept of inbalancedness as described in the introduction and background chapters (Chapters 1,2). Since I aim to demonstrate both the model improvements and general assessment of the new combined test data. Initially, I compare the individual performances of the models as a result of targeted data augmentation designed to create more balanced models. While showing the individual performances I also assess how these models perform against the newly combined balanced test data, which is intended to provide a more comprehensive evaluation of the models.

In order to make it easier to compare the models, both against each other and between the two different test datasets, I will initially provide the results of four models per test dataset. Then, I will interpret the results for each model separately with respect to each test dataset.

5.5.1 Testing against Jenia Test Data (COVID-19)

I compare the performances of the models against the Jenia test data, providing a clearer understanding of how the models perform in relation to previous research.

The provided performance metrics and confusion matrices for the Jenia-10, Jenia-M3, Jenia-M3.1, and Jenia-M3.2 models offer detailed insights into their classification behaviors, especially regarding false positives and false negatives (Table 5.11).

Category	Jenia-10			Jenia-M3			Jenia-M3.1			Jenia-M3.2			Support
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
ADM	0.97	0.47	0.63	**0.98**	0.40	0.57	0.93	**0.64**	**0.76**	0.96	0.52	0.67	891
ATT	**1.00**	0.41	0.58	0.95	0.49	0.64	0.70	**0.54**	0.61	**1.00**	0.49	**0.66**	39
BER	**0.97**	0.27	0.43	0.93	0.12	0.21	0.66	**0.69**	**0.67**	0.81	0.43	0.56	110
ENR	0.97	0.52	0.68	0.97	0.53	0.69	0.90	**0.65**	**0.75**	0.96	0.58	0.72	165
ETN	0.92	0.44	0.60	**0.94**	0.39	0.55	0.78	**0.74**	**0.76**	0.89	0.57	0.69	414
FAC	0.92	0.67	0.78	0.92	0.70	0.80	0.77	**0.79**	0.78	0.89	0.72	**0.80**	281
INS	**0.77**	0.16	0.27	0.73	0.27	0.40	0.54	**0.43**	**0.48**	0.75	0.25	0.38	165
MBW	**0.93**	0.56	0.70	0.85	0.67	0.75	0.81	**0.71**	**0.76**	0.88	0.65	0.75	147
STM	0.78	0.71	**0.75**	**0.88**	0.28	0.43	0.60	**0.83**	0.70	0.71	0.77	0.74	210
Macro Avg	**0.92**	0.52	0.64	0.91	0.49	0.60	0.77	**0.70**	**0.72**	0.88	0.60	0.69	

Table 5.10: Comparison of performance metrics between Jenia-10 (Baseline), Jenia-M3, Jenia-M3.1, and Jenia-M3.2 models against legacy test data

(a) Jenia-10

(b) Jenia-M3

(c) Jenia-M3.1

(d) Jenia-M3.2

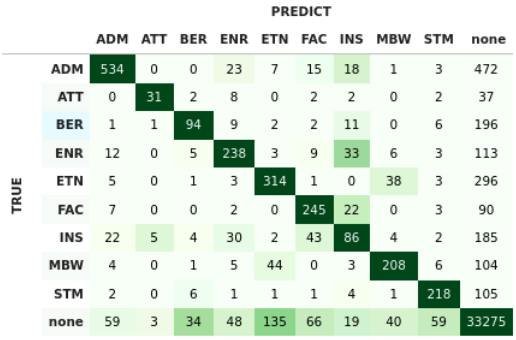
Figure 5.6: Confusion matrices of models tested against Jenia Test data

5.5.2 Testing against Combined Test Data

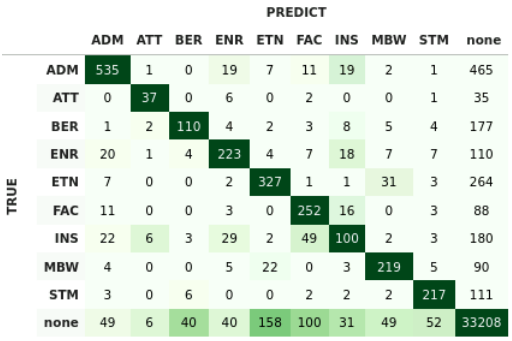
The combined test set integrates data from three different sources: Jenia Test split, Primary Care Dataset of Galjaard (2022), and Oncology Dataset of Badloe (2022). This test set aims to provide a comprehensive evaluation of model performance across diverse datasets, ensuring a robust assessment.

Category	Jenia-10			Jenia-M3			Jenia-M3.1			Jenia-M3.2			Support
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
ADM	0.89	0.50	0.64	**0.90**	0.50	0.65	0.84	**0.66**	**0.73**	0.87	0.55	0.67	1063
ATT	0.91	0.38	0.53	0.86	0.45	0.59	0.70	**0.48**	0.57	**0.92**	0.44	**0.60**	82
BER	0.71	0.28	0.41	**0.73**	0.33	0.46	0.57	**0.60**	**0.59**	0.66	0.40	0.50	331
ENR	0.82	0.58	0.68	**0.83**	0.54	0.65	0.75	**0.68**	**0.71**	0.79	0.62	0.70	413
ETN	**0.67**	0.49	0.57	0.66	0.51	0.58	0.53	**0.77**	**0.63**	0.63	0.61	0.62	1063
FAC	**0.74**	0.68	**0.71**	0.68	0.70	0.69	0.57	**0.80**	0.67	0.70	0.62	0.67	362
INS	0.66	0.23	0.34	**0.70**	0.26	0.38	0.36	**0.48**	**0.41**	0.50	0.31	0.38	186
MBW	**0.82**	0.61	0.70	0.78	0.64	0.71	0.65	**0.75**	0.70	0.77	0.67	**0.72**	341
STM	0.77	0.63	0.69	**0.79**	0.62	**0.70**	0.57	**0.75**	0.65	0.67	0.68	0.67	348
Macro Avg	**0.79**	0.54	0.62	**0.79**	0.55	0.64	0.65	**0.69**	**0.66**	0.74	0.60	0.65	

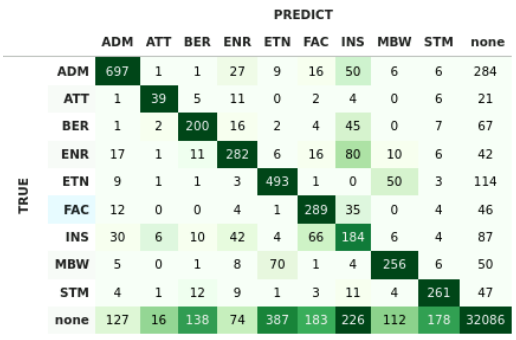
Table 5.11: Comparison of performance metrics between Jenia-10, Jenia-M3, Jenia-M3.1, and Jenia-M3.2 models against combined test data



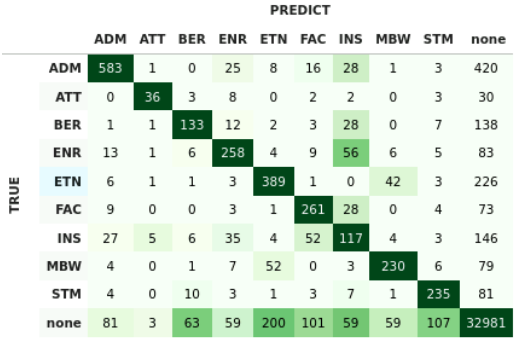
(a) Jenia-10



(b) Jenia-M3



(c) Jenia-M3.1



(d) Jenia-M3.2

Figure 5.7: Confusion matrices of models tested against Combined Test data

5.5.3 Summary of Tests

In Jenia Test data evaluation, Jenia-10 tends to be conservative, resulting in fewer false positives but more false negatives. By conservative, I mean the high precision behavior that the model doe more of a safer classifications which are more likely to be true. Jenia-M3 still faces challenges with both false positives and false negatives. It can be said that it does not perform noticeably different than baseline Jenia-10. Jenia-M3.1 emerges as the best performing model in terms of recall score, effectively reducing the risk of missing true positives (i.e. less false negatives) while maintaining an acceptable level of false positives. If we call Jenia-10 conservative, we can call Jenia-M3.1 more aggressive as it sacrifices some precision in order to catch more instances by takin more risk. These findings suggest that Jenia-M3.1 is the most reliable model, especially in scenarios where capturing true positives is critical. However, since the drop in precision is quite noticable, I also experimented with another AL augmented model which lies in between high precision and high recall. The model Jenia-M3.2 stands as the most balanced improved model overall.

In general all 4 models show relatively worse precision performance on this combined test data. On the top of this, Jenia-M3.1's precision recall trade off brings significantly high false positives on this test data. However, if we observe the confusion patterns in the confusion matrices, we can identify dataset independent classification patterns of the models. The confusion patterns between ETN (eating) and MBW (weight maintenance), INS (exercise tolerance), ADM (breathing), and ENR (energy) remained unchanged with insignificant number differences in each experiment.

Jenia-10

In Jenia Test evaluation, Jenia-10 achieves high precision (0.92) but has a lower recall (0.52) and F1-score (0.64) on average (Table 5.11). This indicates that while the model is quite accurate when it predicts a positive outcome, it tends to miss a significant number of true positives. The confusion matrix for Jenia-10 shows several instances where the model incorrectly predicts positive categories, leading to false positives (5.6a). In the matrix, columns represent the model's predictions while rows represent the true values. For example, ATT was classified correctly 16 times, misclassified as BER 1 times, as ENR 5 times, as FAC 2 times, as INS 1 time, as STM 1 time and finally misclassified as none 13 times. The 'none' misclassifications stands for false negatives. The model has a notable amount of false negatives, as indicated by its lower recall values.

In the combined test data, the performance of Jenia-10 varied more, with generally lower precision and recall scores compared to the legacy test. Categories like ADM saw a decrease in precision to 0.89 and an increase in recall to 0.50, resulting in an F1-score of 0.64. This indicates that the model became slightly better at identifying all ADM instances but at the cost of more false positives. Other categories also experienced a significant drop in performance. These results suggest that the model struggles more with the combined test data, likely due to its increased diversity and complexity.

Jenia-M3

Jenia-M3 demonstrates a balanced precision (0.91) and recall (0.49), with an F1-score of 0.60 (Table 5.11) in Jenia Test set. This balance suggests that the model has a moderate rate of both false positives and false negatives. The confusion matrix for

Jenia-M3 reveals that the model does not behave significantly differently in comparison to Jenia-10 (5.6b).

On the combined test data, the Jenia-M3 model's performance metrics indicate an overall challenge with the more diverse dataset. Most of the categories show a decrease in precision and a slight improvement in recall to 0.50. This shows that the model was able to identify more actual instances but at the cost of higher false positives compared to Jenia-10.

Jenia-M3.1 - High Recall Model

In Jenia Test evaluation, Jenia-M3.1 achieves the highest recall (0.70) and F1-score (0.72) among the three models, with a precision of 0.77 (Table 5.11). This indicates a better balance, capturing more true positives with fewer misses. The confusion matrix for Jenia-M3.1 shows an increase in false positives compared to the other models (5.6c). Although many categories still have misclassifications, Jenia-M3.1 has significantly fewer false negatives, indicating superior recall performance. In every category it has higher true positives compared to Jenia-10 and Jenia-M3. High recall performance comes with the price of relatively lower precision and it brings about more false positives in comparison to other models as it can be seen in the last row (5.6c).

Quite similar to the Jenia Test data evaluation, Jenia-M3.1 achieves the highest recall (0.69) and F1-score (0.66) among the three models again, with a precision of 0.65 (Table 5.11) in the combined test set. The confusion matrix for Jenia-M3.1 shows a reduction in false positives compared to the other models (5.7c). Jenia-M3.1 has noticeably fewer false negatives and higher true positives as result of higher recall performance.

Jenia-M3.2 - Moderate Model

Jenia-M3.2 shows a moderate performance in terms of recall and precision between Jenia-10 and Jenia-M3.1 (Figure 5.6c). It does not outperform any model in any metric, yet its moderate performance still looks the most promising in comparison to Jenia-10 baseline as it shows a significantly better performance without sacrificing precision as much as Jenia-M3.1.

In the combined test set, Jenia-M3.2 shows a moderate performance similarly again in terms of recall and precision between Jenia-10 and Jenia-M3.1 (Figure 5.6c).

5.5.4 Testing against Primary Care and Oncology Datasets

In this section I compare the performance of the moderate model (Jenia-M3.2) against individual constituents of the combined test set. The combined test set enables us to evaluate specific themes in such a modular dataset individually. This comparison gives us more fine grained insight about all model's dropped performance in the combined test set.

Category	Jenia Test Split				Primary Care				Oncology				Combined Test			
	P	**R**	**F1**	**#**	**P**	**R**	**F1**	**#**	**P**	**R**	**F1**	**#**	**P**	**R**	**F1**	**#**
ADM	0.96	0.52	0.67	891	0.59	0.62	0.61	56	0.66	0.77	0.71	116	0.87	0.55	0.67	1063
ATT	1.00	0.49	0.66	39	0.83	0.45	0.59	33	1.00	0.20	0.33	10	0.92	0.44	0.60	82
BER	0.81	0.43	0.56	110	0.60	0.38	0.47	181	0.57	0.42	0.49	40	0.66	0.40	0.50	331
ENR	0.96	0.58	0.72	165	0.70	0.60	0.64	180	0.76	0.79	0.78	68	0.79	0.62	0.70	413
ETN	0.89	0.57	0.69	414	0.29	0.15	0.20	26	0.44	0.75	0.56	200	0.63	0.61	0.62	1063
FAC	0.89	0.72	0.80	281	0.16	0.61	0.25	18	0.55	0.76	0.64	63	0.70	0.62	0.67	362
INS	0.75	0.25	0.38	165	0.25	0.23	0.24	18	0.65	0.48	0.55	101	0.50	0.31	0.38	186
MBW	0.88	0.65	0.75	147	0.80	0.23	0.60	33	0.70	0.74	0.72	161	0.77	0.67	0.72	341
STM	0.71	0.77	0.74	210	0.47	0.41	0.43	49	0.66	0.61	0.63	89	0.67	0.68	0.67	348
Macro Av.	0.88	0.60	0.69		0.57	0.49	0.50		0.70	0.65	0.64		0.74	0.60	0.65	

Table 5.12: Comparison of performance metrics of Jenia-M3.2 against Jenia Test (Kim, 2021), Primary Care (Galjaard, 2022), Oncology (Badloe, 2022)

In the Table 5.12 it can be seen that model M3.2 performs the worst on the Primary Care component of the combined test set. I suppose it would be fair to assume the same for the other models as well. Inferring from low precision and recall, particularly in ETN, FAC and INS categories, it appears that models are particularly unfamiliar with the Primary Care dataset.

5.6 Semi-Supervised Learning Experiment

In this section I compare the Jenia-M3 and Jenia-M3.1 separately with two other respective models created with different fine-tuning data compositions. Jenia-M3 and M3.1 are final models of AL cycle as demonstrated in above experiments. In their data augmentation via AL, I only added low confidence positive instances validated by human experts, discarded the instances turned out to be negative. Here in the model design M3b and M3.1b I also added negative validated instances to the batches.

As a last part of this experiment, I pseudo-labeled in total 16756 high confidence instances and added them on the top of the AL batches. I used my high recall model Jenia-M3.1 while retrieving these pseudo labeled instances.[1] This model augmented with pseudo labeled instances named as M3c and M3.1c.

Pseudo-labeling is a data augmentation technique that Schramm (2023) also adopted in her thesis where predictions are made on selected unlabeled data and these predictions are then used as training instances for the model (Xu et al., 2021). Although I did not focus on this technique in my thesis, I aimed to test whether an Active Learning augmented model, such as Jenia-M3 or Jenia-M3.1, could serve as a better initial model for semi-supervised learning in a small scale.

All of the models are tested against legacy Jenia Test set.

[1] I tried high precision alternative as well, results were not significantly different.

Jenia-M3 vs Jenia-M3b, Jenia-M3c

In this part of the experiment I tested different variations of fine-tuning setup against each other on legacy Jenia test split. In all of these models, I fine tuned *MedroBERTa.nl* with different combinations of datasets using the re-initialization approach as described in the (Section 4.3.2).
Model M3: legacy Jenia Train set + AL Batches 1,2,3 (only low confidence positive instances)
Model M3b: legacy Jenia Train set + AL Batches 1,2,3 (low confidence positive instances combined with annotated negative instances).
Model M3c: legacy Jenia Train set + AL Batches 1,2,3 (low confidence positive instances) + high confidence pseudo labeled positive instances.

Table 5.13: Comparison of different train data constitutions

Category	Precision	Recall	F1-Score	Category	Precision	Recall	F1-Score
ADM-M3	0.98	**0.46**	**0.63**	ADM-M3.1	0.93	0.64	0.76
ADM-M3b	0.97	**0.46**	**0.63**	ADM-M3.1b	**0.98**	0.40	0.57
ADM-M3c	**0.99**	0.41	0.57	ADM-M3.1c	0.75	**0.83**	**0.79**
ATT-M3	0.95	**0.51**	**0.67**	ATT-M3.1	0.70	0.54	0.61
ATT-M3b	**1.00**	0.33	0.50	ATT-M3.1b	**0.95**	0.49	**0.64**
ATT-M3c	**1.00**	0.46	0.63	ATT-M3.1c	0.05	**0.69**	0.10
BER-M3	0.90	**0.34**	**0.49**	BER-M3.1	0.66	0.69	**0.67**
BER-M3b	0.90	**0.34**	**0.49**	BER-M3.1b	**0.93**	0.12	0.21
BER-M3c	**0.91**	0.27	0.42	BER-M3.1c	0.31	**0.85**	0.45
ENR-M3	0.98	0.53	0.69	ENR-M3.1	0.90	**0.65**	**0.76**
ENR-M3b	0.98	**0.56**	**0.72**	ENR-M3.1b	**0.97**	0.53	0.69
ENR-M3c	**0.99**	0.47	0.63	ENR-M3.1c	0.89	0.60	0.72
ETN-M3	0.91	0.48	0.63	ETN-M3.1	0.78	0.74	**0.76**
ETN-M3b	**0.95**	0.43	0.60	ETN-M3.1b	**0.94**	0.39	0.55
ETN-M3c	0.91	**0.50**	**0.64**	ETN-M3.1c	0.39	**0.88**	0.54
FAC-M3	**0.89**	0.68	**0.77**	FAC-M3.1	0.77	**0.79**	**0.78**
FAC-M3b	**0.89**	0.68	**0.77**	FAC-M3.1b	**0.92**	0.70	0.55
FAC-M3c	0.87	**0.69**	**0.77**	FAC-M3.1c	0.33	0.62	0.48
INS-M3	**0.80**	0.25	0.38	INS-M3.1	0.54	0.43	0.48
INS-M3b	**0.80**	**0.31**	**0.45**	INS-M3.1b	**0.73**	0.27	0.40
INS-M3c	0.69	0.16	0.26	INS-M3.1c	0.19	**0.62**	0.29
MBW-M3	0.93	0.57	0.71	MBW-M3.1	0.81	0.71	**0.76**
MBW-M3b	**0.97**	0.56	0.71	MBW-M3.1b	**0.85**	0.67	0.75
MBW-M3c	0.93	**0.64**	**0.76**	MBW-M3.1c	0.35	**0.83**	0.49
STM-M3	**0.81**	**0.71**	**0.76**	STM-M3.1	0.60	**0.83**	**0.70**
STM-M3b	0.81	0.68	0.74	STM-M3.1b	**0.88**	0.28	0.43
STM-M3c	0.81	0.69	0.75	STM-M3.1c	0.19	0.60	0.31
Models	**Macro Pr.**	**Macro R.**	**Macro F1**	**Models**	**Macro Pr.**	**Macro R.**	**Macro F1**
M3	0.91	0.55	**0.67**	M3.1	0.77	0.70	**0.72**
M3b	**0.92**	0.52	0.64	M3.1b	**0.91**	0.49	0.60
M3c	0.91	0.53	0.64	M3.1c	0.44	0.77	0.49

Jenia-M3.1 vs Jenia-M3.1b, Jenia-M3.1c

In this experiment I tested different variations of fine-tuning setup against each other on legacy Jenia test split. In all of these models, I fine tuned *Jenia-10* model with different combinations of datasets using the updating approach as described in the (Section 4.3.2, 4.3.2).
Model M3.1: AL Batches 1,2,3 (only low confidence positive instances)

Model M3b: AL Batches 1,2,3 (low confidence positive instances combined with annotated negative instances)
Model M3c: AL Batches 1,2,3 (low confidence positive instances) + high confidence pseudo labeled positive instances

Summary of Alternative Fine-tuning Data Constitutions

It can be seen in Table 5.13 that adding negative validated instances in Models M3b and M3.1b does not make a significant contribution to performance. The most interesting result of this experiment is the observed inefficiency of pseudo-labeling attempt as it can be seen with the results of M3c and M3.1c models.

5.7 Best Models

Having described the various experiments, in this section I summarize the best models that have been created in the course of the research. In particular, I describe the best models for precision and recall performance, as well as a moderate alterantive between the two.

5.7.1 Best Models for Recall and Precision

The best performing model overall in terms of recall performance appears to be Jenia-M3.1 which is the model created by fine tuning Jenia-10 with only positive data from combined AL batches.

Best Model for Recall

The results in Table 5.14 highlight that Jenia-M3.1 is the best model overall in terms of recall performance. This model, which is fine-tuned from Jenia-10 using only positive data from combined Active Learning (AL) batches, demonstrates superior recall metrics across most categories. Notably, Jenia-M3.1 achieves a macro average recall of 0.70, significantly higher than Jenia-10 (0.46) and Jenia-9 (0.47). This indicates that Jenia-M3.1 is highly effective at identifying true positive instances across various categories, reducing the likelihood of missing relevant data.

Category	Jenia-9			Jenia-10			Jenia-M3.1			Support
	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1	
ADM	**0.99**	0.40	0.57	0.97	0.47	0.63	0.93	**0.64**	**0.76**	891
ATT	**1.00**	0.36	0.53	**1.00**	0.41	0.58	0.70	**0.54**	**0.61**	39
BER	0.80	0.44	0.56	**0.97**	0.27	0.43	0.70	**0.69**	**0.67**	110
ENR	0.95	0.59	0.73	**0.97**	0.52	0.68	0.90	**0.65**	**0.75**	165
ETN	0.94	0.43	0.59	**0.97**	0.44	0.60	0.78	**0.74**	**0.76**	414
FAC	0.90	0.68	0.78	**0.92**	0.67	0.78	0.77	**0.79**	**0.78**	281
INS	**0.77**	0.20	0.32	**0.77**	0.16	0.27	0.54	**0.43**	**0.48**	165
MBW	0.92	0.56	0.69	**0.93**	0.56	0.70	0.81	**0.71**	**0.76**	147
STM	**0.84**	0.66	0.74	0.78	0.72	**0.75**	0.60	**0.83**	0.70	210
Macro Avg	0.91	0.47	0.65	**0.92**	0.46	0.64	0.77	**0.70**	**0.72**	

Table 5.14: Comparison of performance metrics between Jenia-9, Jenia-10, and Jenia-M3.1 models against Jenia Test Data

Best Model for Precision

Table 5.15 identifies Jenia-10 as the model with the highest precision across most categories, outperforming both Jenia-9 and Jenia-M3. Jenia-10 achieves a macro average precision of 0.92, which is negligably better than Jenia-9 (0.91) and Jenia-M3 (0.91). This indicates that Jenia-10 is slightly more accurate in its positive predictions, minimizing the occurrence of false positives. However, it should be noted that it is possible for these models that they are at their higher limits of precision already.

Category	**Jenia-9**			**Jenia-10**			**Jenia-M3**			**Support**
	Prec.	**Recall**	**F1**	**Prec.**	**Recall**	**F1**	**Prec.**	**Recall**	**F1**	
ADM	**0.99**	0.40	0.57	0.97	0.47	**0.63**	0.98	0.40	0.57	891
ATT	**1.00**	0.36	0.53	**1.00**	0.41	0.58	0.95	0.45	**0.59**	39
BER	0.80	0.44	**0.56**	**0.97**	0.27	0.43	0.93	0.12	0.21	110
ENR	0.95	0.59	**0.73**	**0.97**	0.52	0.68	0.97	0.53	0.69	165
ETN	0.94	0.43	0.59	**0.97**	0.44	**0.60**	0.94	0.39	0.55	414
FAC	0.90	0.68	0.78	**0.92**	0.67	0.78	0.92	0.70	**0.80**	281
INS	**0.77**	0.20	0.32	**0.77**	0.16	0.27	0.73	0.27	**0.40**	165
MBW	0.92	0.56	0.69	**0.93**	0.56	0.70	0.85	0.67	**0.75**	147
STM	**0.84**	0.66	0.74	0.78	0.72	**0.75**	0.88	0.28	0.43	210
Macro Avg	0.91	0.47	**0.65**	**0.92**	0.46	0.64	0.91	**0.49**	0.60	

Table 5.15: Comparison of performance metrics between Jenia-9, Jenia-10, and Jenia-M3 models against Jenia Test Data

This result is also an indication that the augmented model via re-initialization (Jenia-M3) could not outperform its initial model (Jenia-10).

Most Balanced Improved Performance

The model Jenia-M3.2 provides a more balanced performance as having relatively greater recall compared to Jenia-10 and M3, and relatively higher precision than Jenia-M3.1 (Table 5.16). This model can be a moderate alternative if the precision drop of M3.1 is found too costly.

Category	**Jenia-9**			**Jenia-10**			**Jenia-M3.2**			**Support**
	Prec.	**Recall**	**F1**	**Prec.**	**Recall**	**F1**	**Prec.**	**Recall**	**F1**	
ADM	**0.99**	0.40	0.57	0.97	0.47	0.63	0.96	**0.52**	**0.67**	1063
ATT	**1.00**	0.36	0.53	**1.00**	0.41	0.58	**1.00**	**0.49**	**0.66**	82
BER	0.80	**0.44**	**0.56**	**0.97**	0.27	0.43	0.81	0.43	**0.56**	331
ENR	0.95	**0.59**	**0.73**	**0.97**	0.52	0.68	0.96	0.58	0.72	413
ETN	0.94	0.43	0.59	**0.97**	0.44	0.60	0.89	**0.57**	**0.69**	1063
FAC	0.90	0.68	0.78	**0.92**	0.67	0.78	0.89	**0.72**	**0.80**	362
INS	**0.77**	0.20	0.32	**0.77**	0.16	0.27	0.75	**0.25**	**0.38**	186
MBW	0.92	0.56	0.69	**0.93**	0.56	0.70	0.88	**0.65**	**0.75**	341
STM	**0.84**	0.66	0.74	0.78	0.72	**0.75**	0.71	**0.77**	0.74	348
Macro Avg	0.91	0.47	0.65	**0.92**	0.46	0.64	0.88	**0.60**	**0.69**	

Table 5.16: Comparison of performance metrics between Jenia-9, Jenia-10, and Jenia-M3 models against Jenia Test Data

Chapter 6

Conclusion, Discussion, and Future Work

In this chapter, I provide a general analysis of the output of this thesis, including error analysis and an interpretation of the overall outcomes of the research. In the following sections, I interpret the outcomes of this research by summarizing the results in their presented order in the thesis (Section 6.1)). Subsequently, I conduct an Error Analysis based on the classification performance of the moderate model Jenia-M3.2 over the combined test set (Section 6.2). Following that, I provide a discussion based on the observations made throughout the thesis, highlighting setbacks, unclear points, and suggesting ways the thesis could have been improved (Section 6.3). Finally, I explore potential expansions of the research presented in this thesis (Section 6.4).

6.1 General Analysis of the Results

In this thesis, my primary goal was to reduce the false positive (FP) and false negative (FN) behavior of the models.

Initially, I addressed two data quality issues within the dataset: misannotations identified as false-false positives in Schramm's (2023) research and misannotations of INS data discovered in Galjaard's study (2022). Correcting these issues increased precision in the evaluations and reduced FPs, providing a more accurate picture of the models' true performance.

Next, I aimed to address the balance of both the test data and the models. For the test data, I combined the legacy Jenia Test split with full datasets from previous domain adaptation research (Galjaard, 2022; Badloe, 2022). This augmented test data provided a more comprehensive evaluation opportunity, as it was thematically divided into three parts, demonstrating that test data can be effectively augmented if needed. It also provided a more balanced category distribution overall.

To balance the models, I employed targeted data augmentation via Active Learning. This methodology showed promising results in addressing model balance and reducing FP and FN behavior overall. I experimented with two approaches to incrementally develop the models. The first approach involved directly fine-tuning the base pre-trained model with a combination of old training data and data retrieved through all AL cycle iterations. This re-initialization approach resulted in the Jenia-M3 model.

In the second approach, I used two ways to incrementally fine-tune the models by continuously updating model weights. For Jenia-M3.1, I fine-tuned a previously

fine-tuned model (Jenia-10) with the newly retrieved data only once. This method significantly improved recall by 20% but sacrificed around 15% precision. I named this model Jenia-M3.1 - High Recall model. In the final method, I continuously fine-tuned the previously fine-tuned model on each iteration, incrementally updating over the previous iteration. This resulted in the Jenia-M3.2 model, which offered a more moderate trade-off, sacrificing only 3% precision for a 10% gain in recall. I included this model in my thesis to provide a balanced option for medical data classification.

High recall is often desirable in medical NLP text classification tasks to capture as many true positives as possible and minimize missed instances. However, a significant drop in precision can lead to unsatisfactory results, as the model would be less reliable for the instances it classifies, despite capturing more instances overall.

This work provides more options for the future development of the A-PROOF project in their use of medical text classifiers, offering a range of models that balance recall and precision based on specific needs.

6.2 Error Analysis

The error analysis is based on predictions over the combined test set with the moderate model Jenia-M-3.2. I wanted to use the moderate model instead of high recall or high precision model because it gives a more balanced error profile between false negatives and false positives. The high precision model produce less false positive, yet more false negatives. Similarly, high recall model makes significantly less false negative predictions but this comes with a price of higher false positive predictions.

6.2.1 False Positives

In this section I showcase representative instances that the model predicted a wrong class. I aim to focus on intra-category confusions when available which I think would provide some insights about disambiguation of these categories.

False positives in model predictions can often indicate overfitting to noise within the data. As shown in Table 6.1, the model incorrectly classified several instances across different datasets, suggesting a tendency to misinterpret irrelevant or misleading features as meaningful. For example, the sentence "85 O2 li 3 6 DEMMI:" from the Jenia Test set was wrongly predicted as ADM, even though its gold value was NONE. The surface pattern which includes some abbreviations related to breathing is likely to be found in breathing specifying instances. Similarly, the simple mention of "Eten/drinken:" was classified as ETN instead of NONE. These type of phrases are probably function as title before the eating/drinking specifying phrase appear. And also I observe that the model is too sensitive towards certain keywords like 'eten'(eating), 'lopen'(walking), 'werkt'(works) where their presence directs the model towards categories like ETN(Eating), FAC(Walking), or BER(Working) while these words might have multiple linguistic functions in different context. These misclassifications highlight how the model can sometimes latch onto superficial patterns or noise, leading to incorrect predictions. This overfitting to noise reduces the model's reliability and underscores the need for further refinement in distinguishing between true signal and irrelevant data.

Sentence	Source Dataset	Prediction	Gold Value
85 O2 li 3 6 DEMMI:	Jenia Test	ADM	NONE
Eten/drinken: *Eating/drinking*	Jenia Test	ETN	NONE
is energie *is energy*	Primary Care	ENR	NONE
eetlust eetlust *appetite appetite*	Oncology	ETN	NONE

Table 6.1: False Positives showing bias towards noise in the data

As it can also be seen in the confusion matrices in the experiments, semantically close categories which are somewhat related to each other in a context are confused a lot by the models. (e.g. ETN(eating) - MBW(weight maintenance), ENR(energy)-INS(exercise tolerance)) When I inspect the false positive misclassifications 6.2, I could again observe surface forms confusing the model. For example, it is likely that the 'reduced nutritional state' confused the model so that it wanted to classify MBW too due to shared vocabulary between MBW and ETN. In the second example in that table, the model misses the implied meaning and focuses on surface form again by focusing on used vocabulary mentions.

Sentence	Source Dataset	Prediction	Gold Value
mevrouw heeft een verminderde voedingstoestand, ziekte gerelateerde ondervoeding zonder inflammatie *The lady has a reduced nutritional state, disease-related malnutrition without inflammation.*	Jenia Test	ETN, MBW	ETN
iets doen (bv. stofzuigen) maar dit kost erg veel energie. Alles *doing something (e.g., vacuuming) but this takes a lot of energy. Everything*	Primary Care	ENR	INS

Table 6.2: False Positives showing confusion between semantically close categories

6.2.2 False Negatives

This section showcase instances that the model predicted as 'None' while in fact they were supposed to belong a category.

Sentence	Source Dataset	Prediction	Gold Value
Vermoeid Gevoel *Tired Feeling*	Ellemijn (Primary Care)	NONE	ENR

Table 6.3: STM False-False Positives from Combined Test Data, Model: Jenia-M3.2

Category	Text Length				Confidence			
	Max	Min	Median	Average	Max	Min	Median	Average
ADM	679.0	2.0	71.0	82.21	0.171119	0.075458	0.118042	0.119485
ATT	285.0	15.0	67.5	85.87	0.173728	0.082028	0.089571	0.097844
BER	452.0	6.0	86.5	103.54	0.184641	0.079614	0.113469	0.121083
ENR	841.0	3.0	74.0	100.56	0.179884	0.084979	0.099287	0.109916
ETN	1805.0	1.0	64.0	99.69	0.195359	0.085369	0.130754	0.132064
FAC	475.0	1.0	57.0	95.71	0.204143	0.085363	0.110774	0.119395
INS	717.0	4.0	79.0	106.08	0.189112	0.080075	0.106352	0.117285
MBW	1051.0	3.0	74.0	111.89	0.190013	0.081700	0.107881	0.117553
STM	739.0	4.0	64.0	96.96	0.183896	0.073680	0.103681	0.116268

Table 6.4: Text length (character) statistics and confidence values for false negative instances per category.

Table 6.4 shows text length and confidence statistics for false negative classifications. While it provides limited insights, the median and average statistics can reveal some aspects of the model's behavior. For example, the relatively high average and median text length statistics for BER, ENR, and MBW suggest that the model has more difficulty classifying these categories in longer sentences. Additionally, the minimum character length for each category indicates that these instances might need re-validation, as it would be challenging to convey any category implication with such short texts.

Regarding the confidence statistics, ETN and BER stand out with their high average confidence in false negative classifications. This suggests that these categories might be more ambiguous compared to those with lower averages, as they tend to have more instances grouped at the lower end with a few high outliers.

6.2.3 New False-False Positives

When I analyze the false positives I suspected some instances should not actually be false positives. I applied False-false positive fixes in (Section 4.1) from an evaluation that I earlier made with the Jenia-10 model on Jenia Test split. However, here I discovered some new instances in this evaluation in the both Jenia Test split, and other datasets of Oncology and Primary Care.

These instances given in the Table 6.5, are singular examples from each dataset from a couple of categories. The medical experts in the A-PROOF team confirmed the presence of a noticeable amount of incorrect false positive predictions throughout the dataset where real gold value should be updated. These findings suggest that the datasets still include notable amount of misannotations where it results with skewed evaluations.

Sentence	Source Dataset	Prediction	Gold Value
Bleef onrustig, toen lorazepam ging werken werd mw rustiger end de band af gedaan *Remained restless, when lorazepam started to work, Mrs. became calmer and the band was taken off*	Jenia Test (COVID-19)	STM (Emotion)	NONE
Maakt zich zorgen dat het niet goed is in het operatie gebied. *Worried that things are not well in the surgical area.*	Oncology	STM (Emotion)	NONE
geweest en schrok hiervan. Houdt *and was shocked by this. holds*	Primary Care	STM (Emotion)	NONE
Kortademig *Short of breath*	Primary Care	ADM (Breathing)	NONE
aanwezig, concentratie verminderd, lezen over tekst heen Bedrijfsarts geeft aan om dan te stoppen. 40 *present, concentration reduced, reading over text Occupational physician suggests stopping then. 40*	Primary Care	ATT (Attention)	NONE
Gewicht 70 kg, stabiel *Weight 70 kg, stable*	Oncology	MBW (Weight Maint.)	NONE
Eet normaal. *Eats normal.*	Oncology	ETN (Eating)	NONE

Table 6.5: STM False-False Positives from Combined Test Data, Model: Jenia-M3.2

6.2.4 Contradictory Predictions - Both Positive and Negative

Logically negative and positive classes should be mutually exclusive. In the sense that no instance can be both none and some other positive category. However, I decided to treat 'none' as just another category alongside the other nine positive ICF categories. The performance of the 'none' category is not 100%, although it is higher than 95%. This decision inevitably leads to some interesting cases of confusion, where certain instances are predicted as both 'none' and another category. The amount is negligible but it is still worth mentioning.

6.3 Discussion

One key point from my findings is the difference between re-initialization and continuous updating in fine-tuning models. While I am not entirely sure what exactly changes during fine-tuning, the difference between these two approaches is understudied (Lemmens and Daelemans, 2023). The re-initialization approach boosted recall but dropped precision significantly. In contrast, continuous updating resulted in a more balanced model. This empirical observation suggests that different fine-tuning compositions can manipulate various performance aspects of the models.

Another important aspect is the trade-off between recall and precision. High recall

is crucial in medical NLP tasks to capture as many true positives as possible. However, this often comes at the cost of precision, leading to more false positives. The challenge is to find a balance that suits the specific needs of the application.

Moreover, I noticed that data augmentation via Active Learning can significantly impact model performance. By carefully selecting informative samples, we can improve model accuracy and robustness. However, this process requires continuous monitoring and validation to ensure that the augmented data truly enhances the model.

At some point, I used a particular query strategy, Cosine Similarity based Redundancy Reduction, but then I dropped it as it turned out not beneficial. This inconsistency in methodology is not ideal for objectively assessing its impact. Ideally, the methodology should be consistent throughout the research. However, applying this would have required additional batches of annotations, which was constrained by time and availability of experts.

Additionally, I included some unintended instances came out of annotations, which were not low-confidence queried instances, into the training batches. Ideally new data consist of only low certainty validated instances. However, I thought they were just extra gold data and would not do harm. They should better be excluded if the experiment could be done with more iterations.

Another consideration is that I used the same data source (VUMC - 2023 Medical Notes) as my unlabeled data pool. It might be beneficial to diversify the source of the unlabeled data pool when possible. This could potentially improve the model's generalizability and robustness.

Overall, these observations highlight the complexities and challenges in fine-tuning and augmenting models for medical NLP tasks. By addressing these points in future work, the methodology can be further refined to produce more reliable and effective models.

6.4 Future Work

Building on the improvements made in this thesis, there are several ways to further enhance the A-PROOF ICF Classifiers Framework.

Further Data Retrieval and Annotation Fixes

The methodology used in this research can be extended to retrieve more data, aiming for a more balanced larger test set. This will ensure that the models are evaluated on a dataset that accurately represents all categories, thus providing a more reliable assessment of their performance.

Both the test data and training data can be augmented with data from different domains to evaluate how these models perform in varied contexts. This will help in understanding the robustness and generalizability of the models.

Experimentation with Query Strategies

Although this was beyond the scope of my background and methodology, I believe Discriminative Active Learning(DAL) should be explored for obtaining more informative samples from the unlabeled data pool (Section 2.4). DAL focuses on training a model to differentiate between labeled and unlabeled data instance types. This binary strategy aims to identify examples that are different from the labeled data, thereby uncovering

new and unexplored data points to improve the model. The work of Gissin and Shalev-Shwartz (2019) [1] can be a useful starting point to discover the potential of this method.

For redundancy elimination, it is worthwhile to try CLS embedding-based techniques, as mentioned in the research by Jacobs et al. (2021). Using the sentence embedding values (CLS) created by the model instead of word embeddings could bring better redundancy reduction performance. These techniques can be combined with various other clustering algorithms such as TF-IDF or K-means to enhance their effectiveness.

Revisiting Sentence Segmentation

The sentence segmentation of the Primary Care dataset can be re-evaluated (Galjaard, 2022). It may be beneficial to segment the data again with specific constraints implemented via a SpaCy pipeline. This could improve data quality by ensuring that sentences are segmented more accurately, leading to better model performance.

Large Scale Bootstrapping

Bootstrapping can be experimented with on a larger scale in combination with AL augmented models. From an initial pool of 750k sentences, approximately 15k positive high-confidence instances were retrieved. It would be worthwhile to scale this approach, perhaps increasing the dataset size by tenfold. Although there is a risk of overfitting, it might enhance the quality and quantity of training data, leading to better model performance.

By exploring these future work directions, the A-PROOF ICF Classifiers Framework can be further refined, resulting in models that are more accurate, robust, and capable of performing well across different domains and conditions. These advancements will contribute to the development of more reliable and efficient medical NLP applications.

In conclusion, this thesis has contributed to improved understanding of medical text classification, through the development and evaluation of improved models and methodologies. By addressing the challenges of data imbalance and enhancing annotation accuracy, this work contributes to more robust and reliable applications of NLP in the medical domain. Future research can build on these findings to further refine models and explore their integration into clinical practice, ultimately contributing to better patient care and outcomes.

[1] https://dsgissin.github.io/DiscriminativeActiveLearning/2018/07/05/DAL.html

Appendix A

Appendix

Table A.1: Train Data Pattern Distribution with Counts and Percentages, Total Pattern Amount 65

pattern	Count	Percentage
O	223198	93.33
ADM	4762	1.99
STM	3262	1.36
FAC	2305	0.96
ETN	2249	0.94
ENR	781	0.33
INS	686	0.29
MBW	610	0.26
BER	445	0.19
ATT	205	0.09
ETN, MBW	99	0.04
FAC, INS	95	0.04
ADM, ENR	75	0.03
ENR, INS	38	0.02
ADM, FAC	34	0.01
ADM, ETN	31	0.01
FAC, STM	25	0.01
ADM, STM	23	0.01
ADM, INS	22	0.01
ENR, STM	19	0.01
ADM, MBW	13	0.01
ATT, STM	13	0.01
ENR, STM	13	0.01
BER, STM	12	0.01
...	...	...

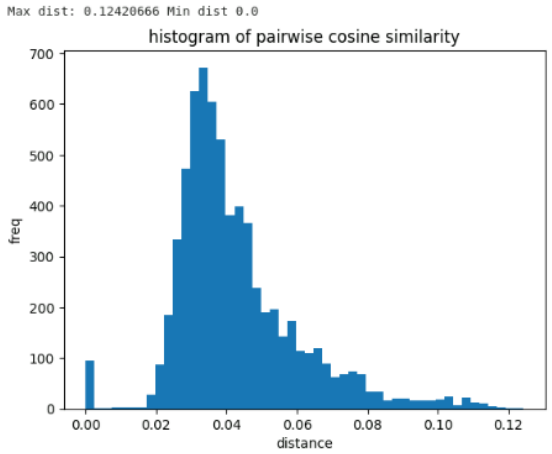


Figure A.1: Pairwise cosine similarity between words from Batch-1

Bibliography

S. Badloe. *Medroberta.nl: Transfer learning from covid-19 to cancer patients. Master's Thesis, Vrije Universiteit Amsterdam.* 2022.

D. Birant and A. Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data Knowledge Engineering*, 60(1):208–221, 2007. ISSN 0169-023X. doi: https://doi.org/10.1016/j.datak.2006.01.013. URL `https://www.sciencedirect.com/science/article/pii/S0169023X06000218`. Intelligent Data Mining.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, 2020.

L. Ein-Dor, A. Halfon, A. Gera, E. Shnarch, L. Dankin, L. Choshen, M. Danilevsky, R. Aharonov, Y. Katz, and N. Slonim. Active Learning for BERT: An Empirical Study. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.638. URL `https://aclanthology.org/2020.emnlp-main.638`.

N. Elhadad, S. Pradhan, S. Gorman, S. Manandhar, W. Chapman, and G. Savova. Semeval-2015 task 14: Analysis of clinical text. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 303–310. Association for Computational Linguistics, 2015.

E. Elyan, C. F. Moreno-García, and C. Jayne. Cdsmote: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification. *Neural Computing and Applications*, 33:2839 – 2851, 2020. URL `https://api.semanticscholar.org/CorpusID:220629130`.

E. H. W. Galjaard. *Evaluating transfer of a functional level classifier from secondary to primary healthcare notes. Master's thesis, Vrije Universiteit Amsterdam.* 2022.

D. Gissin and S. Shalev-Shwartz. Discriminative active learning, 2019.

S. Henning, W. Beluch, A. Fraser, and A. Friedrich. A survey of methods for addressing class imbalance in deep-learning based natural language processing. In A. Vlachos and I. Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 523–540, Dubrovnik,

Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.38. URL https://aclanthology.org/2023.eacl-main.38.

P. Hu, Z. C. Lipton, A. Anandkumar, and D. Ramanan. Active learning with partial feedback, 2019.

P. F. Jacobs, G. M. de Buy Wenniger, M. Wiering, and L. Schomaker. Active learning for reducing labeling effort in text classification tasks, 2021.

D. Jurafsky and J. H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, Upper Saddle River, N.J., 2009. ISBN 9780131873216 0131873210. URL http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210/ref=pd_bxgy_b_img_y.

M. Khushi, K. Shaukat, T. M. Alam, I. Hameed, S. Uddin, S. Luo, X. Yang, and M. C. Reyes. A comparative performance analysis of data resampling methods on imbalance medical data. *IEEE Access*, 9:109960–109975, 2021. doi: 10.1109/ACCESS.2021.3102399.

J. Kim. *A-PROOF Technical Report - Automated Assignment of ICF Functioning Levels to Clinical Notes in Dutch*. 2021.

C. Kuan. *Generative Approach of Data Augmentation for Pre-Trained Clinical NLP System. Master's thesis, Vrije Universiteit Amsterdam*. 2023.

A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6, 2016. doi: 10.1109/CITSM.2016.7577578.

E. Laparra, A. Mascio, S. Velupillai, and T. Miller. A review of recent work in transfer learning and domain adaptation for natural language processing of electronic health records. *Yearbook of Medical Informatics*, 30:239–244, 08 2021. doi: 10.1055/s-0041-1726522.

J. Lemmens and W. Daelemans. Combining active learning and task adaptation with BERT for cost-effective annotation of social media datasets. In J. Barnes, O. De Clercq, and R. Klinger, editors, *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 237–250, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.wassa-1.22. URL https://aclanthology.org/2023.wassa-1.22.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

A. Rawat and S. Singh Samant. Comparative analysis of transformer based models for question answering. In *2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, pages 1–6, 2022. doi: 10.1109/CISCT55310.2022.10046525.

A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349.

C. Schramm. *Using Semi-supervised Learning to Automatically Annotate Dutch Medical Notes for Patients' Functioning Levels. Master's thesis, Vrije Universiteit Amsterdam.* 2023.

C. Schröder, A. Niekler, and M. Potthast. Revisiting uncertainty-based query strategies for active learning with transformers. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.172. URL https://aclanthology.org/2022.findings-acl.172.

B. Settles. Active learning literature survey. 2009. URL https://api.semanticscholar.org/CorpusID:324600.

Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar. Deep active learning for named entity recognition, 2018.

W. F. Styler IV, S. Bethard, S. Finan, M. Palmer, S. Pradhan, P. C. de Groen, et al. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154, 2014.

O. Uzuner. Recognizing obesity and comorbidities in sparse data. *Journal of the American Medical Informatics Association*, 16(4):561–570, 2009.

O. Uzuner, B. R. South, S. Shen, and S. L. DuVall. i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

S. Verkijk and P. Vossen. Medroberta.nl: A language model for dutch electronic health records. *Computational Linguistics in the Netherlands Journal*, 11:141–159, Dec. 2021. URL https://clinjournal.org/clinj/article/view/132.

S. Wankmüller. Introduction to neural transfer learning with transformers for social science text analysis. *Sociological Methods Research*, page 004912412211345, 12 2022. doi: 10.1177/00491241221134527.

Y. Xu, F. Wei, X. Sun, C. Yang, Y. Shen, B. Dai, B. Zhou, and S. Lin. Cross-model pseudo-labeling for semi-supervised action recognition. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2949–2958, 2021. URL https://api.semanticscholar.org/CorpusID:245329496.

X. Yang, J. Bian, W. Hogan, and Y. Wu. Clinical concept extraction using transformers. *Journal of the American Medical Informatics Association : JAMIA*, 2020. doi: 10.1093/jamia/ocaa189.

H. Zhang, H. Zhang, S. Pirbhulal, W. Wu, and V. H. C. D. Albuquerque. Active balancing mechanism for imbalanced medical data in deep learning–based classification models. *ACM Trans. Multimedia Comput. Commun. Appl.*, 16(1s), mar 2020. ISSN 1551-6857. doi: 10.1145/3357253. URL https://doi.org/10.1145/3357253.