Context Attribution To Attain Generalizability in Non-fine-tuned Transformer: A Framework for Concepts Excretion in Cross dataset Evaluation Settings

By

Stepan Tytarenko
BS, Kharkiv National University of Radio Electronics, 2021
MS, Fordham University, 2025

Dr. Ruhul Amin

Computer and Information Sciences Department,
Graduate School of Arts and Sciences

DISSERTATION
SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
IN THE DEPARTMENT OF COMPUTER AND
INFORMATION SCIENCES
AT FORDHAM UNIVERSITY

NEW YORK
DECEMBER, 2024

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's digital age, machine learning models face an increasing challenge: how to effectively adapt to new tasks without requiring extensive retraining. Consider a language model trained to detect fake news about politics - how well would it perform when asked to identify fake news about healthcare? Traditional approaches often struggle with such transitions, requiring significant new training data and computational resources. This challenge highlights a fundamental question in machine learning: how can we build models that truly understand concepts rather than just memorizing patterns?

In the current state of the industry, the dominant strategy for utilizing pre-trained Large Language Models (LLMs) involves fine-tuning with a small subset of task-specific data. While this approach is practical, it often leads to a trade-off between efficiency and performance. Fine-tuning only the model's classification layer is computationally efficient but typically produces suboptimal results compared to full model retraining.

This limitation stems from how traditional fine-tuning approaches modify the model. When only adjusting the final classification layer, we're essentially trying to map complex language representations directly to output categories, without considering the underlying concepts that define these categories. For example, in fake news detection, we're not just looking for specific words or phrases, but rather trying to identify patterns of deception, inconsistency, or bias that characterize fake news across different topics.

This approach offers several key advantages:

- Better generalization across different domains and datasets
- More interpretable decisions, as we can examine which concepts influenced the model's

classification

- Improved performance even with limited training data
- More stable training process

By pre-training the model on a big chunk of semi-structured data and then fine-tuning it with task-specific labeled data, we may obtain state-of-the-art performance in the problems of classification, regression, language translation, and more. The most crucial pre-training stage is usually costly, and repeating it for every new task is computationally inefficient. At the same time, fine-tuning only downstream task-specific heads of pre-trained models is time efficient and requires much less labeled data, preserving models' *generalizability*. On the other hand, this part of the pipeline might be a bottleneck to the process. Usually, single fine-tuning does not produce results on par with the complete model adaptation via training [3]. Explaining and adapting the results to the various downstream tasks is also tricky. To avoid any confusion in this paper, we are going to use the term "training" or "model adaptation", referring to the process of full model retraining (adapting all of the weights), while for the head-only adaptation, we are going to use the term "fine-tuning". Fine-tuning is usually easier and faster and requires less data.

As a potential solution to the problem, we propose a new framework that is used with transformer [4] architecture in order to improve results not only for the specific task but with a significant effect of **generalizibility**. We call this framework a **Context Attribution Model (CAM)** and show that using it instead of a classification layer in transformer models such as BERT [5], DistilBERT [6], XLNet [7], RoBERTa [8] and others is a way of increasing models' performance.

Rather than directly mapping text to categories, CAM introduces an intermediate layer that learns to recognize abstract concepts relevant to the classification task. For instance, when detecting fake news, CAM might learn to identify concepts like "source credibility," "emotional manipulation," or "factual consistency" - concepts that remain relevant regardless of the specific topic being discussed.

A pre-trained LLM can yield reasonable (but not necessarily optimal) natural language

representation, mainly due to the attention mechanism. We assume that representation can be used to perform classification with high accuracy for a set of labels $y^{(i)}$. Since each label holds mutually exclusive concept C_{y^i} for a classification task, we hypothesize that it is possible to learn the concept attributions from the LLM representations of tokens $\vec{e}_k$ for a given document.

The model is designed in a way that a set of concepts describes different classes; such a set is called the "context attribution". It is worth noting that we do not limit these concepts in terms of overlapping. Some context attributions might overlap if that makes sense in terms of the problem solved. This paper reviews one such task where overlapping context attributions are entirely appropriate. What we would like to avoid is allowing multiple concepts to converge to the same representation. For that type of regularization, we introduce an additional loss called **Intra-Space** loss. Its goal is to make sure concepts in the context of attribution are disjointed. Besides, we would like to force concepts to converge to a certain optimum. This optimum is an end goal of training. This requires an additional loss function that we call **Inter-Space** loss. More details on these are given in chapters 5 and 6.

As was stated previously, the Context Attribution Model is an external framework with a set of operators on top of the transformer model. Generally speaking, this is not limited to the transformer architecture either. Potentially, any technique that can produce embeddings may be used as the CAM's base model, such as Word2Vec [9], Glove [10], or RNN [11], [12]. Further in this paper, whenever we refer to the base model, we mean the model that produces the embeddings for the CAM. Some of the base models tested in this paper include BERT [5], DistilBERT [6], and XLNet [7], RoBERTa [8].

The benchmarking and evaluation of the proposed solution are done with various configurations of the base models and across multiple datasets. We test performance for the specific task, fine-tuning the CAM for that particular downstream task, and we also test the performance of the model on the task that is related to the original semantically; however, it uses different data. The baseline for the comparison is mainly the performance of the original base model fine-tuned for the downstream task. During the experiments, we prove that besides an evident performance boost, the Context Attribution Model also stabilizes the training process

and generalizes better for semantically close tasks. We also prove that the model can achieve a significant performance boost even when using a smaller number of parameters than the base model fine-tuned on the downstream task. We empirically show that the performance boost from the smaller number of parameters in CAM is on par with or surpasses the performance of the standard dense layer.

1.1 Role of Generalizability of the LLMs

Currently, the domain of Language Models is one of the most rapidly developing areas of machine learning. Transformer architecture [4] has proven itself as a state-of-the-art approach towards the absolute majority of Natural Language Processing (NLP) domains [13]. A particular strength of the language models is their *generalizability* [14], [15].

This feature is the core strength and the key factor in the popularity of modern Deep Learning Transformers in general and LLMs in particular. There may be different types of generalizability. In Machine Learning recently, the main problem was assuming we have a set of training examples, and we need to train the model to distinguish examples that are similar to these in nature but almost never the same. In that case, the model needs to learn patterns and high-level features that are common to all of the objects across examples. A typical case is when we need to identify human faces. We cannot train the model to recognize every single unique human face. But we can collect a sample that is good enough and train LLM on this sample. Hoping that it will learn these intricacies.

This problem is hard in nature, but we have learned how to solve it with reasonable accuracy for most of the high-demand use cases.

However, there is another view of generalizability. What if we have a set of examples that are examples of patterns for even higher-level patterns? This time, we cannot learn particular features like “human face should have a nose, eyes, etc.”. This time, we are dealing with an entirely different problem each time we interact with the system. And solely based on some high-level concepts, we need to produce outputs. A great example is the ability of large language models to generate coherent text based on diverse prompts. In this scenario, the model

does not rely on predefined features or specific patterns tied to individual tasks. Instead, it leverages abstract representations learned from a vast corpus of data to generalize across tasks and domains. Consider problems like solving math assignments, writing a poem, and answering a question that utilizes facts and knowledge from different domains and fields of knowledge. Such systems align more closely with the concept of "meta-learning" or "few-shot learning", where the ability to generalize depends on recognizing the overarching structure or relationships within the task itself rather than memorizing specific examples. This kind of adaptability is a critical step towards achieving more human-like intelligence in AI systems. A good example of such a task is the recent ARC competition [16]. In this competition, Machine Learning engineers were assigned the task of predicting an output matrix and given a few shots of the example transformations in input-output pairs. According to the authors of the competition, an average human can easily score 85% accuracy on this set of tasks. In contrast, even the most sophisticated automated systems can do no better than 34%. During the competition, the best result that was achieved was 55.5%. This proves how complicated this task indeed is and how crucial and untrivial generalizability is.

1.2 Role of Interpretability of the LLMs embeddings

When working with large language models, particularly with systems that use client data or interact with people, it is crucial to understand the decisions made by the system. Interpretability of the black box model is also a step towards performance improvements. There are multiple common interpretation techniques. Such as [17] and [18]. These are also external methods that use quite similar approaches to interpreting local model predictions of particular examples. Interpretation is achieved mainly through post-hoc, proxy, and feature importance methods.

There are also other more generic approaches [19], researching a question of BERT attention interpretation. Or [20], which tries to build an external tool with the help of attention that could be used to interpret model outputs.

Almost all of the methods add certain assumptions and limitations.

In this work, we focus on interpreting the embeddings produced by LLMs. Since we hy-

pothesize that those correspond to specific concepts, if so, we should be able to identify various concepts with the help of these embeddings. Our assumption is that the external CAM framework produces contextual representations that are well descriptive of the target domain.

1.3 Research Outcomes

The purpose of this paper is to introduce and verify the quality of an external LLM context attribution framework. We first tackle a fake news detection task, utilizing three distinct datasets and outperforming the baseline model in both the same dataset and cross-dataset zero-shot test. Particularly, with our method, we observe improvement by **15%** in terms of accuracy and by **18%** in terms of F1 score on the Covid Fake dataset, **3.5%** of accuracy and **6%** on F1 with the LIAR dataset, **23%** of accuracy and **25%** of F1 on Kaggle Fake News Dataset, and **5%** accuracy and **25%** of F1 on Fake News Net. All results are calculated compared to the BERT model with a fine-tuned classification layer for the fake detection task. In a cross-dataset zero-shot test using fine-tuned Fake News Net dataset to predict Covid Fake data, we observe **5%** accuracy and **25%** F1 boost.

Additionally, experimental results on three datasets, namely HateXplain, IMDB reviews, and Social Media Attributions, illustrate that the proposed model attains superior accuracy and *generalizability*. Specifically, for the non-fine-tuned BERT on the HateXplain dataset, we observe **8%** improvement in accuracy and **10%** improvement in F1-score. Whereas for the IMDB dataset, fine-tuned state-of-the-art XLNet is outperformed by **1%** for both accuracy and F1-score. Furthermore, in an out-of-domain cross-dataset test, DistilBERT fine-tuned on the IMDB dataset in conjunction with the proposed model improves the F1-score on the HateXplain dataset by **7%**. For the Social Media Attributions dataset of YouTube comments, we observe **5.2%** increase in F1-metric.

Furthermore, we introduce a new loss function with accompanying auxiliary metrics, which demonstrates better generalizability, faster convergence, and improved training stability. This new approach also provides insights into how well a dataset represents its domain by measuring generalization across various topics and datasets, supported by a geometric interpretation of

model predictions through embeddings associated with concept definitions (labels).

The impact and novelty of this paper include:

- A novel framework for Language Model fine-tuning, which outperforms the baseline score of the base models such as BERT, DistilBERT, RoBERTa, and XLNet
  - For the non-pre-trained BERT (only trained context attribution operator), we observe **8%** improvement in accuracy and **10%** in F1-score on HateXplain data
  - For the IMDB with XLNet base model, we observe an improvement of around **1%** after full model adaptation compared to fully trained vanilla XLNet
  - CAM with the base model as DistilBERT non-pre-trained (only trained context attribution operator) on IMDB dataset, in a zero-shot manner outperforms basic DistilBERT in the same manner (only head fine-tuning) by **7%** on F1-score
  - Compared to the results from the Social Attribution paper (with manual supervision), we observe an improvement of **5.2%** with our model without additional supervision
  - Demonstration of a significant boost in all metrics for the task of fake news classification
  - Our approach outperforms existing solutions in cross-dataset zero-shot evaluations for Fake detection problem
  - We achieve stabilization of the training process, ensuring consistent and reliable model performance over time
  - The reproducibility of the results is ensured by the code and datasets provided in the repository `https://github.com/StepanTita/space-model`
- A novel loss function that improves the generalization and stabilization of the training process, improving the zero-shot capacity of the transformers

Chapter 2

Background

Before diving into the technical details of our approach, it's important to understand the key concepts and technologies that form its foundation. This chapter provides an overview of transformer models, attention mechanisms, and the challenges of model fine-tuning.

Consider how humans process language: when reading a sentence, we naturally pay attention to different words depending on the context and our task. For example, in the sentence "The movie was great but the popcorn was stale," we focus on different parts depending on whether we're interested in the film review or the concession stand experience. Transformer models attempt to replicate this ability through their attention mechanism.

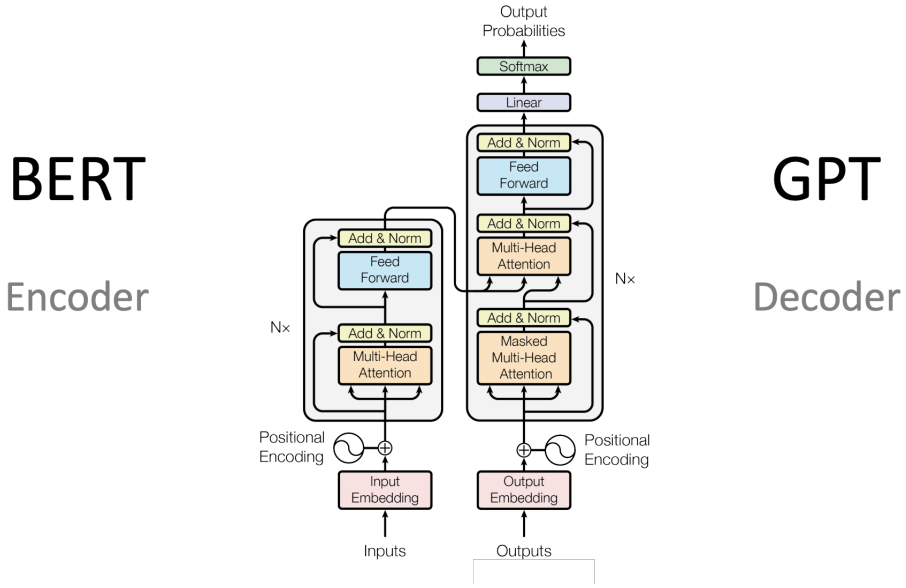

Figure 2.1: Transformers architecture from [1].

Transformer models are the backbone of the LLM revolution. The architecture was first introduced in the 2017 Google Paper [1] and has since become the standard for building LLMs. The key feature of transformer models is the attention mechanism. This mechanism serves multiple purposes:

- It allows the model to focus on different parts of the input sequence
- It acts as the model's memory, retaining information from previous inputs
- It enables the model to make connections between different parts of the sequence for making predictions

Transformer models revolutionized natural language processing by introducing several key innovations besides attention:

- **Parallel Processing**: Unlike previous sequential models, transformers can process all words in a sentence simultaneously
- **Attention Mechanism**: Allows the model to weigh the importance of different words differently for each prediction
- **Positional Encoding**: Maintains word order information without requiring sequential processing
- **Multi-head Attention**: Enables the model to capture different types of relationships between words

2.1 Overview of BERT

BERT (Bidirectional Encoder Representations from Transformers) is a transformer model that was first introduced in the paper [5]. BERT is an Encoder-Decoder model, meaning that it has two main components: an encoder and a decoder. Models that only have an encoder are usually called autoencoders, while models that only have a decoder are usually called seq2seq models. Key features of BERT are:

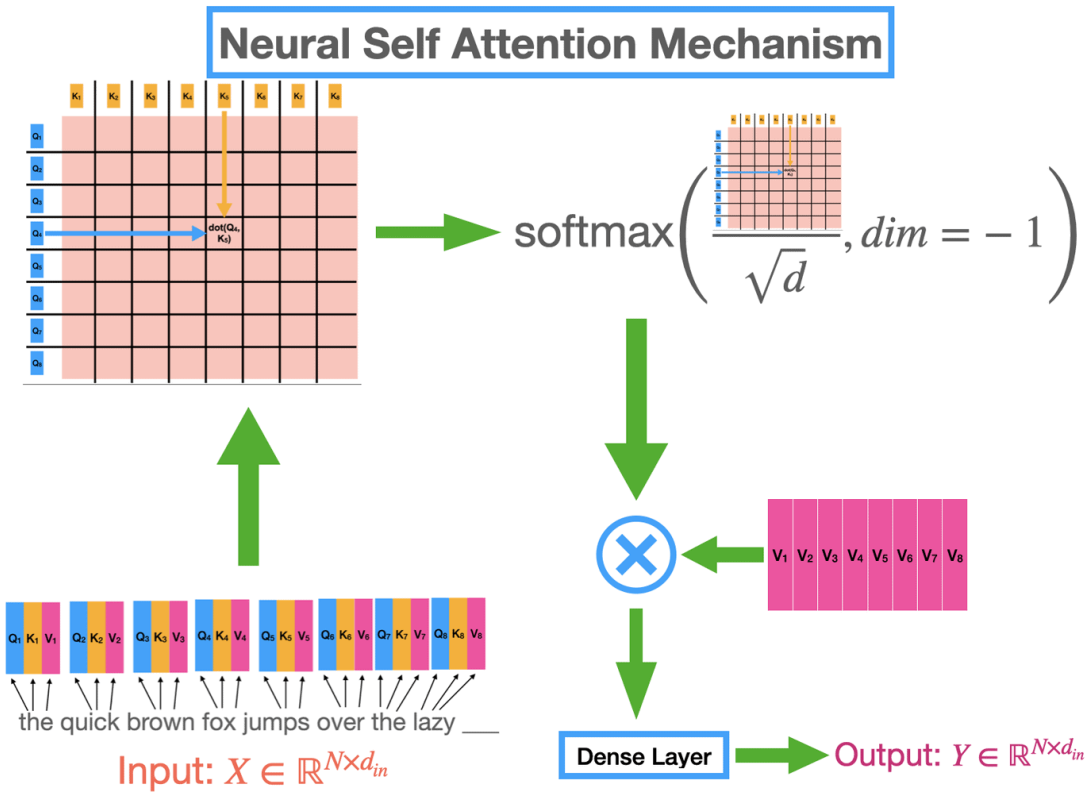

Figure 2.2: Self Attention Mechanism from [1].

- Bidirectional Transformer: BERT is built upon the transformer architecture but with a bidirectional transformer instead of the usual left-to-right transformer.
- Pre-training: BERT is pre-trained on a large corpus of text and then fine-tuned for specific tasks.
- Contextualized Representations: BERT learns to represent words based on the context in which they appear, allowing it to capture nuances in language.

2.2 BERT Pipeline

2.2.1 Tokenization

Transformers, just as any other LLM, uses tokenization to split the input text into smaller units. The most common tokenization method is to split the text into subword units. Different models use different tokenization methods, but the most common ones are Byte Pair Encoding (BPE) and WordPiece. Besides basic tokens, there are also special tokens that are used to indicate the

beginning and end of a sentence, as well as to handle padding and other tasks. These special tokens are:

- `[CLS]`: Indicates the beginning of a sentence.
- `[SEP]`: Indicates the end of a sentence.
- `[PAD]`: Used to pad the input to the same length as the other inputs.

`[CLS]` is used very important in the BERT pipeline, as its embedding is used to represent the whole sentence. This is used in the classification tasks, where the entire input sequence is used to make a prediction.

2.3 Sentence Transformers

Sentence Transformers [21] is a type of transformer model that is specifically designed to work with sentences. They are typically used for tasks that involve processing sentences, such as sentiment analysis, semantic similarity, and text retrieval.

In this paper, we are not using Sentence Transformers directly; however, we will use ideas from the paper.

2.4 Model Fine Tuning for Text Classification

The key task of this paper is to fine-tune a pre-trained transformer model for text classification. There are many different ways to fine-tune a model, but in this paper, we focus on two of the most common ones:

- Full Training: The model is fine-tuned on the task by training the entire model on the task-specific dataset.
- Downstream Head Fine-Tuning: The model is fine-tuned by training only the downstream head of the model while keeping the hidden parameters frozen.

2.4.1 Vector Embeddings representations from LLM

Vector embeddings are a way to represent text as vectors. They are used in many different tasks, such as clustering, classification, and retrieval. Vector embeddings can be generated in many different ways, but in this paper, we focus on using the hidden states of a pre-trained transformer model. The idea is to take the output of the transformer model, which are the hidden states of the last layer, and to use those as vector embeddings. This has the advantage of already capturing the semantics of the text and that the embeddings are in the same space as the model's weights, which can be used for downstream tasks. We are particularly interested in this approach because we need to create context attribution maps for the model's predictions.

2.5 Text classification Large Language Models

2.5.1 DistilBERT

DistilBERT [6] is a distilled version of BERT, meaning that it is a smaller model with fewer parameters but with close to the same capabilities as BERT. The main idea of DistilBERT is to use knowledge distillation to create a smaller model that is still able to match the performance of BERT. Knowledge distillation is a technique where a smaller model is trained to mimic the behavior of a larger model. This is done by training the smaller model on the same dataset as the larger model but with the output of the larger model as the target. DistilBERT uses the same tokenization method as BERT, and the same pre-training procedure, but with some minor modifications to the architecture. We can assume that DistilBERT is just a smaller version of BERT and that (to some extent) the results for DistilBERT are also applicable to BERT.

2.5.2 RoBERTa

RoBERTa [8] is a variant of BERT that was introduced in 2019. The main idea behind RoBERTa is that the improvements of BERT can be accredited to the training procedure, and not to the model architecture. Therefore, RoBERTa uses the same model architecture as BERT but with a different training procedure. RoBERTa has proven to be even better than BERT for many tasks.

2.5.3 XLNet

XLNet [22] is a transformer-based model introduced in 2019 that aims to improve upon BERT's limitations. Unlike BERT, which uses a masked language model (MLM) objective and assumes independence between masked token predictions, XLNet employs a permutation-based training objective. This approach enables XLNet to capture the bidirectional context in a way that models dependencies between all tokens in a sentence. XLNet is a current state-of-the-art model for many NLP tasks.

Chapter 3

Literature Review

The task of effective fine-tuning is one of the main tasks in the modern NLP. Cheaper and faster results of great quality are very appealing and a current trend in the domain. However, we are sourcing inspiration for our framework not only from the latest NLP findings but also from other sources. The core idea has a root in a Psychological Belief Attribution Theory [23], [24]. The theory revolves around the idea of attribution of certain concepts with corresponding behavior patterns. The concepts (sometimes also referred to as factors) may be external and internal. These factors are usually related to personal beliefs, and they affect the decisions and behavior of an individual. Researchers have also classified people based on these factors (e.g., pessimistic attribution, optimistic attribution, hostile attribution). We try to apply the same idea to language modeling, attributing certain concepts with class labels. In general, the idea of measuring and researching the belief attribution of language models is not novel. The authors of [25] have not only proved that certain language models possess the beliefs, but they have also provided metrics to measure such beliefs and a tool to update these beliefs, as well as visualization of beliefs graph.

It is very natural that semi-supervised solutions are mentioned when it comes to fine-tuning with the least resources. These also mainly include ensembling to achieve regularization when working with unsupervised data. One of the first such approaches addressing this issue is the COREG [26]. The technique uses two k-nearest-neighbor regressors with different distance metrics to label the data. The distance metric, in that case, would serve as the confidence label.

This approach uses a fundamental idea that some features in some spaces are aligned with similar class labels and are further apart from the different class labels. This is an essential fact that is reused in the CAM.

Another later technique involves minimal supervision for the labeling of the concept space, and then, based on this concept space, the model can autonomously label the unlabelled data [27]. The key idea here is the knowledge extraction from the manually labeled concept space. It is claimed in the work that labeling a set of concepts and then running an algorithm on a set of documents to label them based on these supervised concepts is a superior technique. Our main takeaway from there is that we can extract knowledge from the supervised concept space for unlabelled data. Furthermore, what we would like to propose is testing if this concept space can help us make a prediction at the inference stage rather than during labeling.

The Social Media Attributions paper [28] addresses a task similar to ours. However, their approach differs in two key ways. First, it requires supervised sub-concept labeling. Second, they measure similarities between sub-concepts and model attention by feeding each sentence multiple times with different sub-concepts. However, they are using the similarity measure to find the concept sub-space that best describes the given sentence and make the decision based on that. Our approach does this all in one pass and in an automated manner. We do not require manual labeling of the concept sub-spaces. We expect to learn them during the fine-tuning phase.

In the paper on Interacting Conceptual Spaces [29], the authors create all of the necessary mathematical background required to formulate the knowledge extraction process from the concept space. They converge the optimization task to the convex relations and prove that by optimizing the conceptual space and merging multiple concepts (or even spaces) together, one can extract new knowledge practical for the downstream task. They also provide an algebra language on how concepts are organized and interact and what it means mathematically when several concepts (or concept spaces) are combined. They put the conceptual representations in different compacts and explore the vectors' behavior there. This is one of the ideas we are adopting in our paper, which we believe helps regularize the network. The concept spaces are

encapsulated into a compact hypercube with the side 2. This is achieved due to the utilization of the $tanh$ activation, which we will review in more detail in the methodology section.

A study on deep learning performance improvement by Erion et al. [30] refers to the concept of attribution in a slightly different manner. They refer to the feature attribution. The hypothesis is that some features that are noisy or redundant may get more attention than others. To deal with this issue and simplify the interpretation of the results, authors introduce attribution priors to optimize for higher-level properties of explanations, such as smoothness and sparsity, enabled by a fast new attribution method formulation called expected gradients that satisfies interpretability axioms.

Recent work on language model tuning [31] proposes an entirely external proxy framework that can be fine-tuned separately without the involvement of original model weights. The outputs (logits) of the initial model are then added to the logits of the fine-tuned proxy model to get optimized output. That way, authors preserve the original capabilities and save resources on model adaptation while also claiming some good performance results. This approach resembles around the same idea that we explore in our paper.

A recent study on embedding space interpretation [32] is motivated by the idea of enabling humans to understand the vector space of fine-tuned models. In order to achieve that, authors hypothesize that they could create some alternative projection of the original latent representations into comprehensible conceptual space. The core idea is very similar to what is being researched in the current work. The authors propose to have a few stage projections. First of all, they obtain latent representations from the model by embedding the original text. These embeddings transform with specific external operators. Besides, they also create representations of certain concepts, and then, by applying arbitrary similarity measures, they attribute input text to particular predefined concepts. The key difference from our approach here is that we do not assume any predefined concepts; rather, we encourage the model to explore alternative contextual representations, and later, we explore if these can be interpreted. In the original paper, the authors built conceptual spaces based on the Wikipedia category-directed graph as ontology.

Chapter 4

Datasets

4.1 Datasets Overview

For this research, we utilize four different datasets of fake news 4.1. Three of them are binary labels: Fake COVID [33], Fake News Kaggle [34], FakeNewsNet [35], and one contains multiple labels LIAR (multi-label) [36]. We also use 4 other datasets for different types of benchmarking: Hatexplain [2], GoEmotions [37], IMDB movie reviews [38], Social Media Attributions [28]. So, we prove that our model evaluates better in both binary and multi-label approaches across different domains.

The datasets used for benchmarking are HateXplain [39], IMDB reviews sentiment dataset [40], and Social Media Attributions dataset of YouTube comments, related to Chennai water crisis [28]. The main reason for choosing corresponding datasets is that HateXplain is considered a very complex dataset, with imbalanced data, and labels "offensive" and "hateful" are conceptually very close. On the other hand, IMDB sentiment reviews are a semantically close dataset to the former one and are reasonably easily interpretable. Such a relation is essential since we would like to test the *generalizability* of the proposed technique. Besides, in the Social Media Attributions paper, the authors apply a very similar approach to the one proposed in this paper, however, with additional manual labeling of the concepts and multiple runs. We would like to show that our approach achieves superior performance without additional manual labeling and via a single pass.

We also use the GoEmotions dataset to measure how well we can interpret the embeddings produced by our framework.

Dataset
Fake COVID [33]
LIAR (multi-label) [36]
LIAR (binary-label)
Fake News Kaggle [34]
FakeNewsNet [35]
HateXplain [2]
IMDB [38]
GoEmotions [37]
Social Media Attributions [28]

Table 4.1: Benchmarks Tested

Fake news detection benchmarking is not a novel idea; moreover, it is a rapidly developing field of research. The International Fake News Detection benchmark [41] provides a notable set of datasets for model evaluation. They analyze both the image and textual parts of the datasets, focusing on comparing different models within each dataset. In contrast, in this work, we focus on the textual part, testing the same model across datasets.

4.2 HateXplain

Model	Text	Label
Human Annotator	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
CNN-GRU	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
BiRNN	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
BiRNN-Attn	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
BiRNN-HateXplain	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
BERT	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	OF
BERT-HateXplain	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	OF

Figure 4.1: HateXplain dataset Rationale Labels. Green highlights tokens found important by both models and human annotators. Orange - words that models find important but human annotators did not. [2].

HateXplain [2] is a dataset of hate speech and offensive language. The final dataset is composed of 9,055 posts from Twitter and 11,093 posts from Gab. The dataset is split into 3 classes: hate speech, offensive language, and normal language.

4.3 IMDB Movie Reviews

The IMDB Movie Reviews dataset [38] is a dataset of movie reviews from the Internet Movie Database. The dataset contains 50,000 reviews, split into 25,000 reviews for training and 25,000 reviews for testing. The reviews are split into 2 classes: positive and negative. We use it in conjunction with the HateXplain dataset to test if the model is able to generalize to domains other than social media.

4.4 Social Media Attributions

Social Media Attributions [28] is a dataset of YouTube comments about the 2019 Chennai water crisis. The dataset contains 72,098 comments from 43,859 users on 623 videos related to the crisis. The comments are annotated for attribution ties - factors that users hold responsible for the crisis (such as poor city planning or population growth). The dataset includes 2,500 annotated comments that can be used to study how people attribute blame for collective misfortunes through social media discussions.

4.5 The COVID-19 Fake News

The COVID-19 fake news dataset was introduced in [33], and it contains Twitter posts with some fake and truthful facts about COVID-19 or related topics. It is worth noting that this dataset has a particular domain and differs from a dataset focusing on political news. However, our research shows that our model scales well across multiple domains of knowledge by grasping intrinsic high-level concepts of what is fake and what is non-fake.

4.6 The LIAR

The LIAR dataset was introduced in [36] and contains short statements in various contexts from PolitiFact.com. The important distinction of this dataset is that the target variable is non-binary. We tested it in both scenarios and showed that our models scaled well for both cases.

4.7 Fake News Kaggle

Fake News Kaggle Dataset is a dataset from kaggle competition [34]. We show that without full model retraining and any hyperparameters optimization, our CAM Model fine-tuned on this data would not only rank in the top 8 of this competition but would also be well generalized for the cross-dataset evaluation.

4.8 Fake News Net

Fake News Net is a dataset developed by [35] and containing news from GossipCop and PolitiFact. Those are in the same domain but with different data sources. So we also check that using only the GossipCop part of the data, the model would *generalize* well for the PolitiFact data.

4.9 GoEmotions

GoEmotions [37] is a dataset of Reddit comments annotated for 27 emotions. We use it to see if our proposed context attribution maps are able to capture the emotional content of the text.

4.10 Dataset Quality Analysis

An important observation is that with the CAM approach, we can measure how good the dataset itself is. When we train the model using only the GossipCop part of the data and then perform zero-shot cross-dataset evaluation, we observe better results than after training on GossipCop and PolitiFact combined. This tells us that the latter dataset has some underlying issues and should be used cautiously.

Chapter 5

Methodology and Experiments

5.1 Problem Definition

Consider a large language model that takes a document $d = [t_1, t_2, ..., t_T] = [t_i]$, i.e., a sequence of tokens of length T, as input and produces a sequence of contextual embedding for each token in N-dimensional vector space, R^N. We can define such language model as: $E = LLM([t_i]) = (\vec{e}_1, \vec{e}_2, \vec{e}_3, ..., \vec{e}_{n_d})$, where $\vec{e}_k$ is in R^N.

Any categorical label is a concept that can have multiple attributes representing it. For example, the label "fake" can be attributed to terms or a set of concepts that represent or contextualize the fake pieces of news. On that idea, we extrapolate the categorical label $y^{(i)}$ as a spectrum for several concept attribution terms (i.e., synonyms representing the concept), $C_{y^{(i)}} = c_1, c_2, c_3, \ldots, c_{m_C}$, such that; $\forall i \exists j \ : \ ATTR(C^{(j)}, y^{(i)})$, meaning - each categorical label has at least one concept attribution term. Thus a given classification task can be expressed as, $\tau = \tau_1, \tau_2, \tau_3, \ldots, \tau_{l_\tau}$, comprising subtask, t_i for each label $y^{(i)}$, of attributing the document $[d]$ by learning the attribution transformation function:

$$ATTROPT(T_{y^{(i)}}, LLM([d])) \rightarrow \text{document classification}$$

where $T_{y^{(i)}} = M_{|C_{y^{(i)}}| \times N}$ (some learnable matrix of given dimensionality, specific features of this matrix are discussed further).

$$C_{y^{(i)}} = ATTR(E, T_{y^{(i)}})$$

Based on these assumptions we define the following optimization task: there exists a learnable transformation tensor, $T_{y^{(i)}}$, for each label $y^{(i)}$ such that it can operate on $LLM([t_i])$ token representations $\vec{e}_k$ for a given document $[d]$ to project those tokens on to an abstract concept space. The tensor $T_{y^{(i)}}$ can be optimized using customized loss functions during the training for a classification task. Vectors in the concept space can be used as the attributions in the downstream tasks to improve the discriminator function's performance and provide *generalizability* for the classification of $LLM([t_i])$.

5.2 Algorithm proposal

We find projection of all document tokens, collectively defined as E, by applying attribution transformation operator, $C_{y^{(i)}} = ATTR(E, T_{y^{(i)}})$, where $E_{T\times n}, T_{n\times|C_{y^{(i)}}|} \implies P_{T\times|C_{y^{(i)}}|}$. $P_{y^{(i)}}$ - is the projection of E using learned tensor $T_{y^{(i)}}$ The abstract concept space for each label $y^{(i)}$ as $R^{|C_{y^{(i)}}|}$ such that the transformation for each token $\vec{e}_k$ would be a vector $\vec{p}_{y^{(i)}}$. As our hypothesis states, to use the concept attribution vectors in the subspace $R^{|C_{y^{(i)}}|}$ for the downstream tasks, we apply the $tanh$ normalization function. This helps us to scale the subspace vector components within a bounded region of $[-1, +1]$. Thus, each token in $LLM([t_i])$ will have a concept space vector $\vec{p}_{y^{(i)}}$ in $R^{|C_{y^{(i)}}|}$, where each component of the vector represents the attribution score for the respective concept.

5.3 Training objective

As defined in the previous section, we use auxiliary losses in conjunction with the primary Cross Entropy Loss function. The final form of an objective function is as follows:

$$loss = CrossEntropyLoss(X, y|\theta) + \lambda_1 J_1(P_{y^{(i)}}, y|\theta) + \lambda_2 J_2(P_{y^{(i)}}|\theta)$$

Where λ_1 and λ_2 are the weights of the auxiliary losses. Setting λ_2 to high values would

mean more robust regularization of the intra-space loss. This loss ensures that the concepts describing the concept space do not converge into a single representation. Setting λ_1 to higher values would make inter-space loss dominant compared to Cross Entropy. This type of behavior is unique to each specific use case. In the experiment section, we show in some instances, optimizing Concept Spaces representation also optimizes the target logits. However, this might only sometimes be the case.

We provide an illustration to describe the architecture and show which particular parts are used for the loss evaluation in fig 5.1.

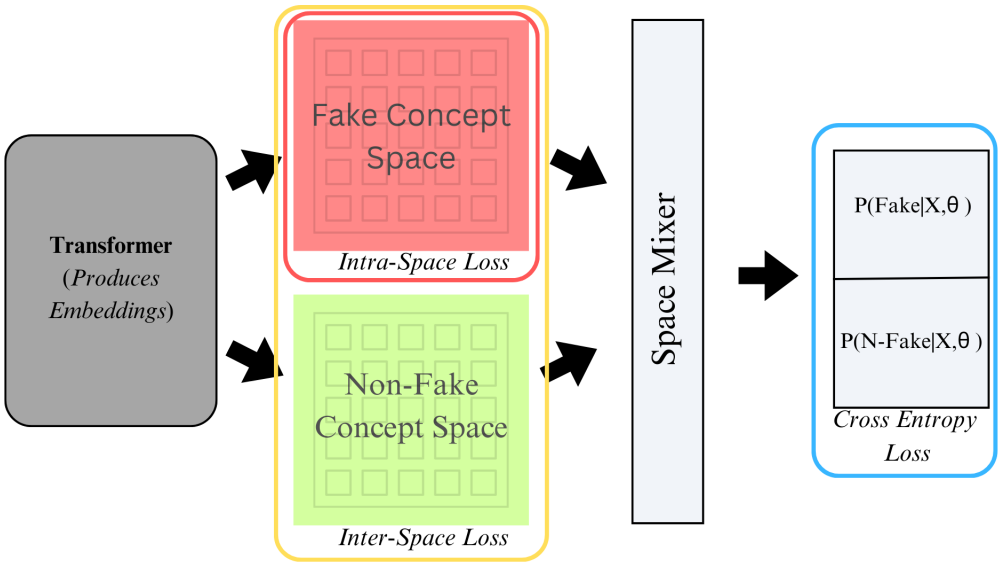

Figure 5.1: CAM-Model diagram with specifying outputs used for loss components computation

5.3.1 Attribution function and Auxiliary losses

The transformation function, $ATTR(E, T_{y^{(i)}})$ is computed as follows:

- **Linear Transformation**: for each attribution for the concept, $C_{y^{(i)}} = c_1, c_2, c_3, \ldots, c_{m_C}$ we perform a dot product with all tokens resulting in, $P_{y^{(i)}} = E \times T_{y^{(i)}}$
- **Normalization Constraints**: As our hypothesis states, to use the concept attribution vectors in the subspace $R^{|C_{y^{(i)}}|}$ for the downstream tasks, we apply the $tanh$ normalization function. This helps us to scale the subspace vector components within a bounded region

of $[-1, +1]$. Thus, we define:

$$ATTR(E, T_{y^{(i)}}) = tanh[\forall_{j,k}(P_{y^{(i)}})] = tanh[\forall_{j,k}(E \times T_{y^{(i)}})]$$

Thus, each token in $LLM([d])$ will have a concept space vector $\vec{p}_{y^{(i)}}$ in $R^{|C_{y^{(i)}}|}$, where each component of the vector represents the attribution score for the respective concept to the token.

- **Regularization And Generalization Terms**: We define these terms to avoid overfitting and to achieve *generalizability* as follows:

$$w(\theta) = \frac{1}{|C_{y^{(i)}}| \cdot T} \sum_{\forall_{j,t}} tanh[P_{j,t}])$$

$$J_1(\theta) = (L \cdot [\sum_{k=1}^{|C_{y^{(i)}}|} (1 - w(\theta)] + (1 - L) \cdot [\sum_{k=1}^{|C_{y^{(i)}}|} (1 + w(\theta))]$$

Where L is a vector of labels where each label is in 1 to 1 correspondence with the number of concept spaces.

$$J_2(\theta) = \sum_{k=1}^{|C_{y^{(i)}}|} \frac{1}{\sum_{c_j \in C_{y^k}}^{n} Var[c_j]}$$

5.3.2 Primary Loss function

The loss that we are optimizing is primarily the Cross-Entropy loss. To ensure that the conceptual embeddings don't converge to the same embedding inside the conceptual space, we introduce an intra-space loss J_2. This also adds additional regularization and improves *generalization*. This is proved during the experiments. Controlling the weight of this loss compared to the cross entropy loss is another hyperparameter fine-tuning task. The intra-space loss is basically an inverse of the variance of the vectors inside the context attribution. To make sure that embeddings converge towards the particular context - we introduce inter-space J_1 loss.

5.4 Auxiliary metrics CS Accuracy and CS F1

The main goal of $J_1(\theta)$ is to ensure that the projections on correct and incorrect concept spaces are highly polarized. Specifically, we would want that when the embeddings are projected on the correct concept space (fake concept space for the fake piece of news), we would like that representation to approach certain polarization. When this same fake embedding is projected onto the non-fake concept space, we would like this polarization to be orthogonal.

This is achievable due to the defined $tanh$ normalization that we use in the concept space projection. Based on the loss definition and the fact that the values of the projection dimensions are bounded between -1 and 1, we state that correct representations will have a representation close to the point $[1, 1, ..., 1]$, while incorrect representations will look like $[-1, -1, ..., -1]$ (this is exactly what the definition of the inter-space loss tells us). If that is the case, then the mean of the correct representation would also approach 1 for that case (note, by "correct" here, we mean the one that matches the concept space assigned with a label to the underlying true label of the embeddings that is being processed).

With that, we have a criteria of label assigning. After projecting embeddings on the concept spaces, take the one that has the highest mean of the representation vectors. This also means that the produced context attribution is closely attributed to this corresponding concept space or label. Formally:

$$pred_{cam} = \underset{i \in |L|}{\operatorname{argmax}}[\frac{1}{T \cdot |C_{y^{(i)}}|} \sum_{t,k}^{T,|C_{y^{(i)}}|} C_{y^i}[t,k]]$$

These metrics are designed to explain why the final model with auxiliary losses works well and *generalizes* much better. Training this loss not only gives some excellent results in both the same dataset and cross-dataset tests but also converges at much higher rates. This metric is a good measurement of improved *generalizability*. That means that by just attributing the context, we capture the intricate structure of the problem rather than the structure of the particular dataset.

Previously, we introduced the Auxiliary Metrics, which helped us describe and evaluate our model's performance. The reason these metrics are optimized is based on the definition of

the loss function. Let us define the optimization goal for the inter-space loss $J_1(\theta)$, introduced in the previous section. In the model, we train transformation tensor $T_{y^{(i)}}$ for each label $y^{(i)}$. Each label is assigned a unique concept space that would attribute the context of this label. For instance, the "fake" concept is assigned to one concept space, and the "non-fake" concept is assigned to another concept space 5.1 (note that in the general case, we can have arbitrarily many labels, each assigned with specific concept space, attributing its context).

This model aims to extend the existing architecture, so after we have attributed the contexts with some tensor transformation, we want the model to make a decision taking both attributions into account. For that, we introduce the Space Mixer layer. It combines the output of the concepts and learns to attribute given text to the specific label. However, we hypothesize that a decision can be made solely based on the context attribution of the token embeddings on each concept space.

5.5 Evaluation Metrics

Since we are evaluating the model between multiple benchmarks simultaneously, we want to adjust to both a perfectly balanced IMDB dataset and a less balanced HateXplain dataset. So, we report accuracy and f1-macro score. Our loss throughout the experiments is cross-entropy loss combined with intra-space loss and inter-space loss for better regularization.

The interpretation of the CS Accuracy and CS F1 is exactly the same as for the standard Accuracy and F1, only calculation changes. Besides, for the explainability part, we will need to introduce multiple distance metrics and reformulate some existing metrics to better understand the shape of the underlying concept space. Since, in this case, we work with embedding vectors, we first of all have to get the proximity of those vectors. We will be extracting the top 1, 3, and 5 candidates in a simple kNN manner. We embed the concept [42] (standard natural language definition of the label), and each of the returned vectors is also an embedding. For the scenario when $k > 1$, we take the mean of whatever metric is applied to vectors. For instance, we use Jaccard Similarity; when the top 3 or 5 results are returned, we measure the similarity between each returned result and the embedding of a label definition and report the average of

the results. Same for Mean Cosine similarity and Mean Euclidean Distance. We also report Min cosine similarity and Max Euclidean distance among the returned results. This helps to understand when foreign concepts are extracted. For example, we use the sentence expressing aggression and extract aggression, anxiety, and joy. The first two are clearly related concepts, while the third one is clearly foreign.

5.6 Context Attribution

Context attribution - is a projection of a collection of contextual embeddings (vectors) or, simply, a matrix. Projection is done via a concept operator. When we train the model, we ensure that concept operators project disjoint concepts far away from similar concepts. We project the sentence in multiple context attributions and then find the similarity between the original sentence and the conceptual projections. This similarity tells how to classify the instance correctly. Note that in the actual implementation, we do not do the pairwise comparisons of the similarities or any other type of processing. Instead, we concatenate obtained projections into a single tensor and feed it to the classification layer. Thus, instead of manually defining the classification criteria, we specify it as a set of trainable parameters.

5.6.1 Contextual word embeddings

As was already stated, the only assumption we impose on the base model is that it can create (contextual) embeddings from the input. Let N_s be the sequence length, and d be the dimensionality of the contextualized embedding of the model. $E = [e_1, e_2, ..., e_{N_s}] \in R^{N_S \times d}$, $E_{N_s \times d} \in R^{N_S \times d}$ is the contexual embedding matrix.

In our research, we assume that the BERT-like models produce this embedding. So N_s would be defined in the range between 256 and 512 (as a maximum sequence length used by the BERT architecture), and d would be 768 for all of the base models and 1024 for the XLNet large.

5.6.2 Conceptual projections

For each of the classes in the classification problem, we assume a single concept space operator. This concept space operator transforms (projects) the contextual embeddings to the context attribution and produces new conceptual embeddings. The obtained representation of the embeddings is also called a latent representation. This representation's dimensionality is defined as the latent space (target space for the projection). Let m be the dimensionality of the latent space. We first define the projection operator as a matrix with trainable parameters: $P_{d\times m} \in R^{d\times m}$. Thus obtained projection matrix (context attribution) $C_{N_s\times m} \in R^{N_s\times m}$.

Basically, context attribution is a new representation of the embeddings in the latent space, where the transformation operator is trained during fine-tuning. However, since we want to obtain proximity of the contextualized embedding to the context attribution, we introduce previously defined $tanh$ operation as a similarity measure.

$$C_{N_s\times m} = tanh(E_{N_s\times d} \times P_{d\times m}) \tag{5.1}$$

$tanh$ is applied element-wise. In that case, our conceptual matrix is a representation of how close a certain sentence is to the concept from the target context attribution (1 is very close, and -1 is from an orthogonal attribution). As an example, when we feed the word "terrible" to the context attribution that was predefined as "positive", we expect to see -1 in the conceptual representation and 1 for a word like "great".

The training objective of the model is to take into account multiple projections of the input embeddings to find the projection that is most aligned with the sentence content. The similarity measure we are using is a slight modification of the cosine similarity, where normalizing the value by the vectors' norms is replaced with the $tanh$.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{5.2}$$

Similarly to the original paper introducing LSTM [43] , we use $tanh$ to control the flow of the information in the network. It squashes the range, centers the values around zero, and in-

troduces non-linearity. This has also proven to be an excellent regularization technique, which reduces the instability, improves models' *generalizability*, and improves the results. This aspect is discussed in more detail in the results section.

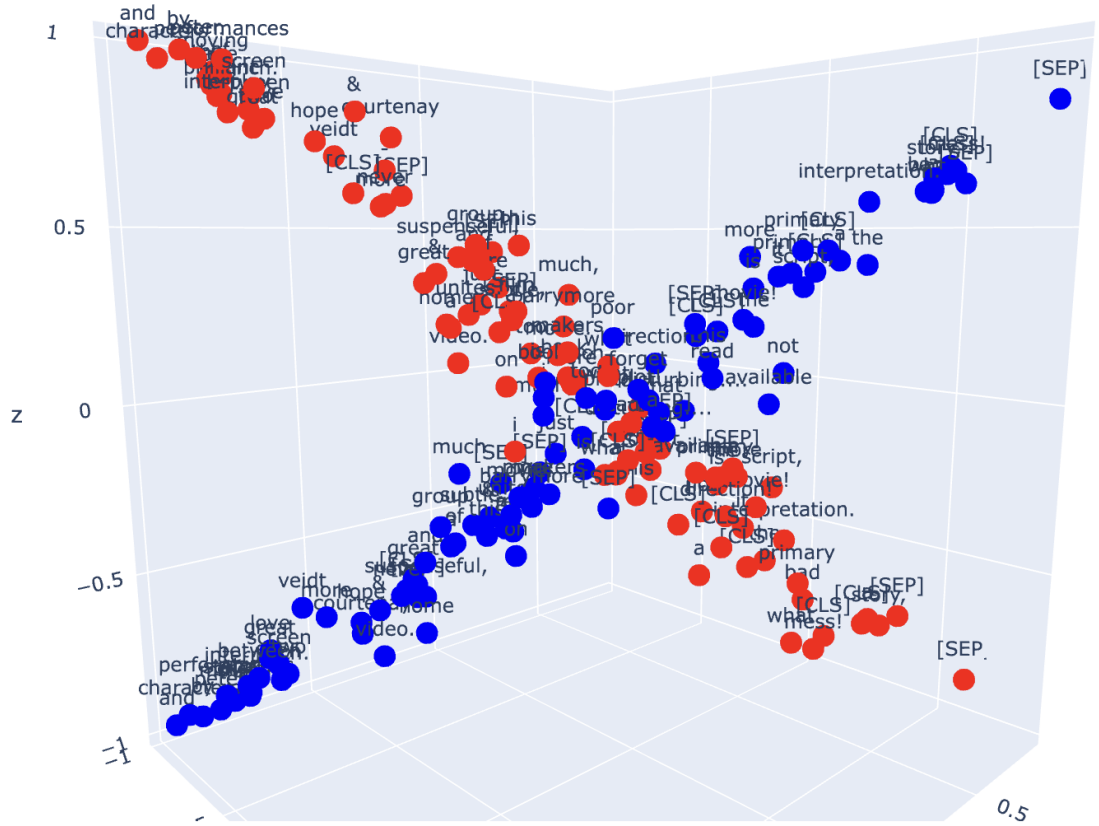

Figure 5.2: 3D projection of the space embeddings for the 2-class classification

As a good side-effect of the $tanh$, we add additional non-linearity and squashing effect to the model. Thus, no additional normalization of values is required. Besides, we shrink our problem to the compact (hypercube with side 2, from -1 to 1). In 5.2, one can find a benefit from such an approach. We can now easily interpret the outcome of the binary classification model. The visualization provided is the contextual embedding of the BERT model into 3-dimensional context attribution space for the IMDB classification task (this is done for test examples, so the model is not overfitted, and what we clearly see is the orthogonality of the negative and positive sentiment concepts).

According to our definition, every target class has a unique context attribution for itself. So for n classes classification problem:

$$C^i_{N_s \times m} = tanh(E_{N_s \times d} \times P^i_{d \times m}) \tag{5.3}$$

for $i \leq n$. Where C^i and P^i are namely i-th context attribution and concept projection operator.

5.7 Vectors' similarity

Vectors' similarity is one of the key concepts in this research. Basically, the entire idea of contextual embedding revolves around the vectors' similarity. In this paper, we primarily focus on cosine similarity. Or, more accurately, dot product, which is unnormalized cosine similarity.

5.7.1 Cosine Similarities

Cosine similarity measures the cosine of the angle between two non-zero vectors in an inner product space. It is particularly effective in high-dimensional, sparse data scenarios. The general formula for cosine similarity is:

$$cos(A, B) = \frac{A \cdot B}{|A| \cdot |B|}$$

This measure ranges from -1 (opposite directions) to 1 (same direction), with 0 indicating orthogonality.

In our research, we replace the normalization step with non-linearity such as $tanh$. But the logic remains the same.

5.7.2 Nearest Neighbors

The nearest neighbor concept identifies the closest data point(s) in a dataset to a given query vector. In our case, we first of all use it to interpret the results. We embed the corpus of the concepts, and then instead of doing standard classification - we find k nearest neighbors of the embedded text among the embedded concepts.

5.8 Measuring Concepts Explainability and Interpretation

As part of the framework proposal, we want to see if our model helps us better understand the decisions made by the model. And if we can benefit from this interpretation in any way. For that, We would have a set of various distinct concepts. Some of them might be very close in meaning; some might be the opposite. We will have a dataset labeled with these concepts. Using our learning operator, we will learn a model to distinguish between these concepts. Later, we hypothesize building a knowledge space by providing just a concept definition, embedding it, and finding the Euclidean (or cosine) distance between the definition and text embedding. In this space, close concepts are closer to each other; thus, sentences representing some of these concepts would also fall within this cluster of concepts. This would help us understand the spatial structure of the concept spaces that the model is generating. If we see that a certain sentence is closer to the specific definition embedding than to the others - this sentence is more likely to belong to that concept. If this hypothesis holds for the data, then we know that concept spaces indeed attribute the context to the conceptual level, providing more semantical understanding rather than pure contextual meaning.

5.9 Classification

After we have projected the embeddings to all of the context attributions, we need to perform classification. In that case, since every projection is a set of vectors (where each vector is a conceptual embedding with latent size m), we would find the centroid of this representation for each context attribution and then concatenate these representations. This concatenated representation is then fed to the single linear layer for classification. This basically identifies the proximity of the embedding to the corresponding context attribution. Let k_i represent i-th context attribution centroid, and $c_{i,j}$ j-th conceptual embedding vector (j-column) of the i-th context attribution $C^i_{N_s \times m}$.

$$k_i = \frac{1}{N_s} \cdot \sum_{j=0}^{N_s} c_{i,j} \tag{5.4}$$

5.10 Theoretical Analysis of CAM Framework

The superior performance of the CAM framework can be attributed to several theoretical advantages:

5.10.1 Geometric Interpretation

The CAM framework's effectiveness stems from its geometric properties in the embedding space. When we project embeddings onto context attributions, we create a structured representation where:

- Each context attribution defines a subspace that captures task-specific features
- The orthogonality between different context attributions ensures a clear separation between classes
- The projection operation acts as a form of regularization, preventing overfitting to dataset-specific features

5.10.2 Information Bottleneck Perspective

From an information theory standpoint, CAM implements a form of the Information Bottleneck principle:

$$I(X;T) - \beta I(T;Y) \tag{5.5}$$

Where:

- X represents the input text
- T represents the context attribution projection
- Y represents the target classification
- β balances the compression-relevance trade-off

This formulation helps explain why CAM achieves better generalization: it retains only the most relevant information for the classification task while discarding noise and dataset-specific artifacts.

5.10.3 Loss Function Analysis

The effectiveness of our auxiliary losses can be understood through their complementary roles:

- **Inter-space Loss**: Promotes orthogonality between different context attributions, ensuring each captures unique aspects of the classification task
- **Intra-space Loss**: Prevents collapse of embeddings within each context attribution, maintaining rich representational capacity

The combination of these losses creates a well-structured latent space that facilitates better generalization across datasets.

5.10.4 Relationship to Attention Mechanisms

CAM's context attribution can be viewed as a specialized form of attention that operates at a higher level of abstraction than traditional transformer attention:

- While transformer attention focuses on token-level relationships, context attribution captures concept-level relationships
- This hierarchical structure allows CAM to learn more robust and transferable features
- The projection operation effectively filters out dataset-specific noise while preserving task-relevant information

Chapter 6

Results

We evaluate our framework using 4 base models with 8 different datasets. With benchmarks, we want to measure:

- Performance of the proposed CAM Framework
- Generalization property of the novel technique
- How does it compare with existing context attribution solutions which involve a manual process

To achieve that, we are going to use 8 datasets: FakeNews (of different kinds), HateXplain, IMDB reviews sentiment dataset, and Social Media Attributions dataset of YouTube comments related to the Chennai water crisis. IMDB sentiment reviews are a semantically close dataset to the HateXplain and are reasonably easily interpretable. Such a relation is essential since we would like to test the *generalizability* of the proposed technique. This same logic applies to the multiple Fake Datasets that we've introduced before. We want to prove that the model learns high-level patterns and relations and is able to use them to make better predictions. Besides, in the Social Media Attributions paper, the authors do manual labeling of the concepts and measure similarity with the so-called Social Media Attributions; we would like to show that our approach achieves superior performance without additional manual labeling and via a single pass.

The experiments are structured in a way that we have basic experiments with smaller models

and simpler tasks. Additionally, we conducted experiments to compare the CAM to the state-of-the-art model of the IMDB dataset. We also investigate and analyze various properties of the CAM and explore some of the hyperparameters' usage, with their respectful effect on the model performance. We explore the *generalization* property of the model by cross-testing it on the unseen dataset.

6.1 Preprocessing and settings

BERT serves as a standard reference and baseline model in our experiments. Its weights are fully trained only once when evaluated against the Chennai water crisis dataset. For all other datasets, we preserve the model's *generalizability*, avoiding additional training. Similarly, XL-Net, as the state-of-the-art transformer for multiple benchmarks, is evaluated on the IMDB sentiment analysis dataset. In this context, we aim to demonstrate that attaching the CAM head to any modern transformer can significantly enhance performance.

For both datasets, we conduct minimal preprocessing. The only exception is for the Fake News Kaggle Competition dataset, where we combine the title and text fields. All experiments employ cased models, except in the Social Media Attributions comparison. We utilize cased DistilBERT, cased BERT, and cased XLNet as base models in our framework, which focuses on generating contextual embeddings. These base models are flexible and interchangeable, depending on the use case.

The CAM model incorporates a latent space whose dimensionality is a hyperparameter. Experiments are conducted with values of 3, 64, 128, 256, and 512 dimensions, with 64 dimensions providing an optimal balance between complexity and efficiency for most of the experiments. Training is conducted for 50 epochs, using the Adam optimizer with a learning rate of $2 \cdot 10^{-5}$, a batch size of 256, and a maximum sequence length of 512. The joint loss for model optimization includes three components, with inter-space loss weights typically set between 0.1 and 0.01, and intra-space loss weights between 10^{-5} and 10^{-7}. These weights are normalized to prevent dominance from any single loss component.

As a rule of thumb, intra-space loss is estimated by calculating the variance for a few sam-

ples and averaging across the batch size using a non-fine-tuned model. This provides guidance for proper weight setting, regularization, and avoiding dominance. Notably, the inter-space loss often converges in 10-20% of the epochs required for Cross-Entropy loss convergence, as demonstrated in our ablation study.

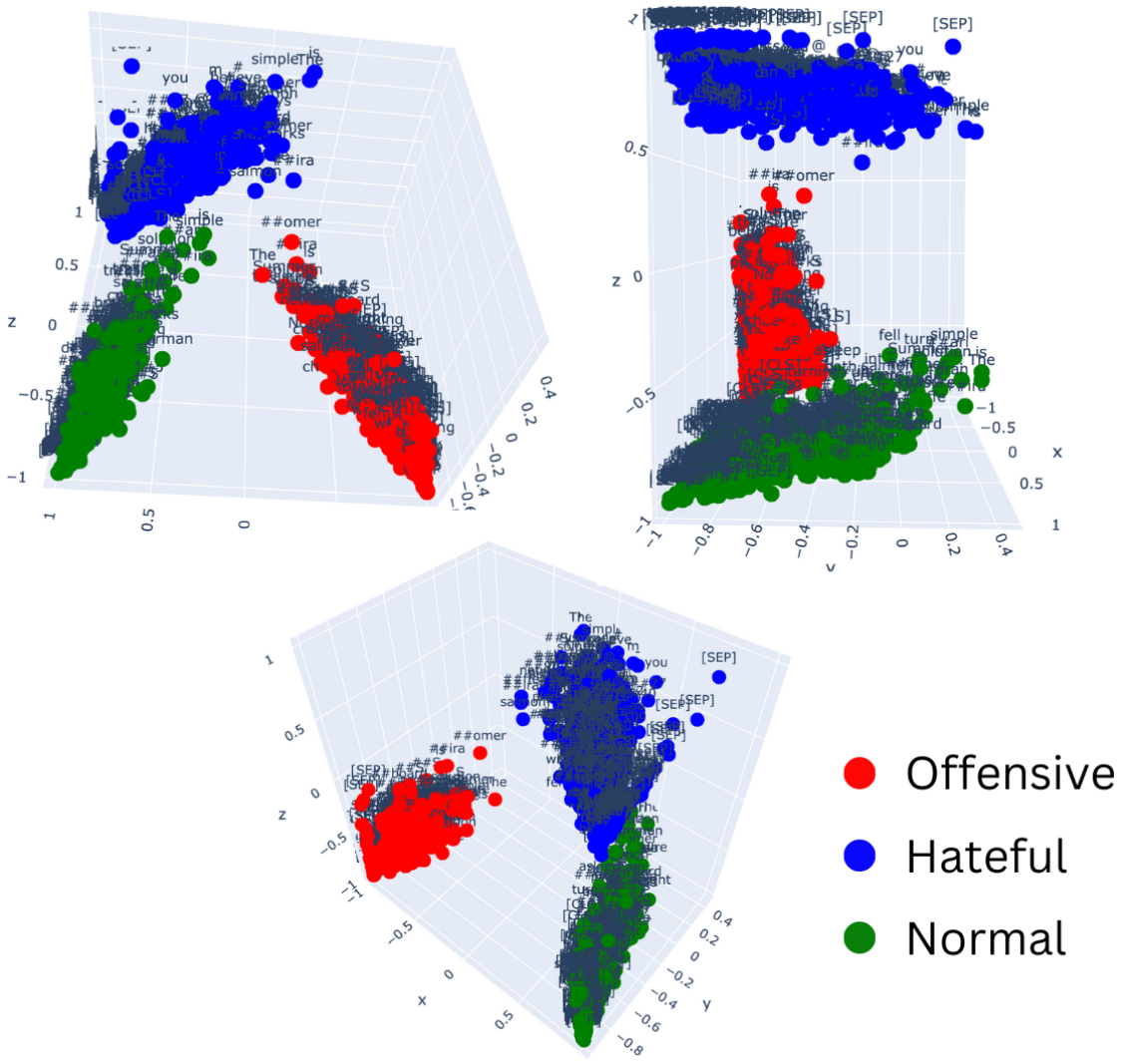

Figure 6.1: 3D projection of the space embeddings for the 3-class classification (HateXplain)

We use the base model configuration for all of the experiments except for the state-of-the-art establishment (12 layers for BERT and XLNet and 6 layers for DistilBERT). For the state-of-the-art performance, we trained large (24 layers) XLNet, which was used for reporting the results in the original paper. We use the Adam optimizer with a learning rate of $2 \cdot 10^{-4}$ for all experiments, except for the state-of-the-art establishment, since the original paper states that 10^{-5} was used to achieve the best results. Maximum sequence length and batch size is 256 for

all of the basic experiments and is replaced with 512 and 4, respectively, for the XLNet large state-of-the-art results.

Since the original paper recommends using 32 as the batch size for the IMDB benchmark for the best results, and we could not fit that to the GPU memory, we used 8 gradient accumulation steps and adjusted the number of training steps accordingly. Even though the result does not precisely reproduce the original outcome, it is close, and the evident performance boost from the CAM is transparent.

For the number of latent spaces, we use three for most experiments since this is enough to outperform significantly and is easy to visualize. As discussed previously, when we project the contextual embeddings onto the context attribution, we expect these projections to be orthogonal if the classes are different. That is what we observe in Figure 1 and Figure 2.

For the comparison with the Social Media Attributions, use the latent size of 64. For the state-of-the-art results using XLNet, we use 128 as the latent space size. We use a single Nvidia A5000 GPU for our training. Our model with various configurations may take from 30 seconds per epoch with DistilBERT to 25 minutes with XLNet large. A standard number of fine-tuning epochs is set to 5; however, for the XLNet large state-of-the-art results, we used only one epoch of training with one epoch of head fine-tuning to prevent overfitting.

Due to the nature of our novel loss, we also introduce additional metrics: Concept Space (CS) Accuracy and Concept Space (CS) F1. This is evaluated as accuracy and F1-score, where the label is chosen based on the $tanh$ normalization between concept spaces (the label corresponding to the closest concept space is assigned) 5.1. After projecting the transformer embedding onto the concept spaces, we obtain two latent representations. On the means of these representation vectors, we apply $tanh$ to obtain normalized values, and based on the concept of proximity, space (label) is assigned.

6.2 Experimental Results

We prove the efficiency of our novel framework via multiple benchmarks. Our data comes from different sources, such as Twitter, political websites, news websites, and more. We use

the datasets provided in the table 4.1 for the evaluation. All of our evaluations are based on the BERT transformer architecture. We train the model on different data and test it through different domains (e.g., training on political news and testing on COVID-related news). Since our framework only utilizes the outcome of the transformer network, replacing BERT with a superior model and using Space conceptual attribution on top of that would achieve an even better performance boost (we show that our framework is model agnostic using RoBERTa and DistilBERT in some tests).

Our primary objective in this work is to show that our novel loss function with the novel architecture performs better than the standard model (BERT in this case) with the same and cross datasets. With this approach, we prove that we do not just learn the specifics of the dataset but also make the model understand the high-level concepts of fake vs. non-fake.

With CAM-Model fine-tuning, we achieve performance close to the entire model training but with far less memory consumption during training and without the loss in *generalizability*.

6.2.1 CAM-Framework vs. Dense Layer

Ultimately, our main contribution is reimagining the final downstream task head, specifically for the classification tasks. To prove that using the framework as the final layer is beneficial in terms of generalizability and just for the results, we compare it to the dense layer, which has a much bigger or comparable number of parameters. The results are reported for the DistilBERT on two datasets: IMDB Movie reviews [38] and Fake news. We observe that performance depends on the task and on the model. For the IMDB dataset, we keep a number of CAM-model parameters at 2.6% of those in the Dense layer. While for fake datasets, we boost that number to 65%.

Model	**Dataset**	**# Params**	**Accuracy**	**F-1**	**Precision**	**Recall**
DistilBERT	fake	592,130	0.7970	0.7945	0.8031	0.7939
CAM-DistilBERT	fake	385,002	**0.8243**	**0.8220**	**0.8317**	**0.8210**
DistilBERT	IMDb	592,130	0.7890	0.7890	0.7890	0.7890
CAM-DistilBERT	IMDb	15,402	**0.8211**	**0.8211**	**0.8212**	**0.8211**

Table 6.1: Comparison of models and their performance metrics

Dataset	Model	# Params	Accuracy	F1	Recall
IMDB (training)	DistilBERT	592130	0.7852	0.7819	0.6614
	CAM DistilBERT	197122	**0.8322**	**0.8320**	**0.8663**
HateXplain (zero-shot)	DistilBERT	592130	**0.6013**	0.4450	0.0869
	CAM DistilBERT	197122	0.5821	**0.5187**	**0.2698**

Table 6.2: Comparative table of the results of the CAM in different configurations with DistilBERT on the IMDB dataset and HateXplain dataset

6.3 Generalizability

We take corresponding models and evaluate them on the HateXplain benchmark in a zero-shot manner 6.2. For the sentiment analysis, negative labels are encoded as 0 and positive as 1; for the HateXplain, we encode Hateful and offensive labels as 0 and normal labels as 1. Here, we see that the CAM with intra-space loss is a top model in terms of f1-score, while the accuracy is the highest for the DistilBERT. However, accounting for the dataset imbalance, we see that DistilBERT is worse in terms of f1 by at least 6-7% and almost 4 times worse in terms of recall. Next, we compare the BERT model with the CAM-BERT and base BERT on the HateXplain benchmark with 3 classes. Here, we only train the classification head for BERT and contextual attribution operators for the CAM framework (as discussed previously). The results in 6.3 clearly show that the CAM is superior in all of the metrics by at least 8%.

Metric	CAM-BERT	BERT
Accuracy	**0.5296**	0.4485
F1-score (macro)	**0.4304**	0.3314
Precision	**0.5431**	0.4471
Recall	**0.5296**	0.4485

Table 6.3: BERT HateXplain (3-class) evaluation

We then fine-tune the classification head and the CAM with XLNet and BERT base models for the same HateXplain benchmark 6.4 and observe that for BERT, the performance gap with identical training settings and identical base model is more than 16% on the f1-macro score. In comparison, for XLNet, this gap is around 6%.

To further prove the effect of the performance boost using the CAM framework, we do the full training of the XLNet, a state-of-the-art model for the IMDB benchmark 6.5. With almost identical settings to the original paper, we obtain a 0.9386 f1-score, while training the CAM

Metric	# Params	Accuracy	F1	Precision	Recall
CAM-XLNet	4622	**0.8798**	**0.8797**	**0.8764**	**0.8824**
XLNet	1538	0.8160	0.8156	0.8421	0.7750
CAM-BERT	4622	**0.8110**	**0.8108**	**0.8227**	**0.7899**
BERT	1538	0.6588	0.6555	0.6919	0.5649

Table 6.4: BERT and XLNet Comparison on HateXplain Dataset (2-class)

with the exact same settings gives us **0.9487** (all of the other metrics are also superior for the CAM, except for the recall, which is again very different with precision for the vanilla model, and very close for the proposed approach). We observe that the CAM framework surpasses the state-of-the-art models in the tasks and is much more tolerant of imbalanced data. The precision-recall trade-off is evident in most of the experiments. To prove this point further, we conducted the ablation study and researched how the framework stabilizes performance during training.

Metric	CAM-XLNet	XLNet
Accuracy	**0.9488**	0.9387
F1-score (macro)	**0.9487**	0.9386
Precision	**0.9463**	0.9106
Recall	0.9516	**0.9731**

Table 6.5: State-of-the-art XLNet on IMDB

6.3.1 Invidividual Dataset Evaluation, CAM-BERT vs BERT

After fine-tuning the default BERT and various configurations of CAM-BERT on all of the datasets we select as benchmarks, we compare them in the table 6.6. This experiment set proves that our model is superior if used for the same dataset. We also claim it is superior and generalizes much better than fine-tuned BERT.

6.3.2 Cross-dataset Evaluation, CAM-BERT vs BERT

We define labels in the following manner: 1 - stands for fake for all of the datasets, and 0 - encodes true. By doing that, we ensure that the evaluation is consistent so that we can test it in a zero-shot manner. For the cross-dataset case 6.7, we see that the best-performing model is the one with 128 latent dimensions. It could capture the most context attributions of the fake and

Data	Architecture	# Dim.	Accuracy	CS Accuracy	F1	CS F1
Fake COVID News	BERT	N/A	0.7145	N/A	0.6966	N/A
	CAM-BERT	3	0.7949	0.5234	0.7828	0.3436
	CAM-BERT	64	0.8645	**0.7743**	0.8627	**0.7637**
	CAM-BERT	128	**0.8808**	0.6528	**0.8797**	0.5967
Liar (multi-label)	BERT	N/A	0.2221	N/A	0.1211	N/A
	CAM-BERT	3	0.2362	0.1824	0.1540	0.1079
	CAM-BERT	64	**0.2580**	**0.2362**	0.2034	**0.1590**
	CAM-BERT	128	0.2572	0.2081	**0.2120**	0.1267
Liar (binary-label)	BERT	N/A	0.5666	N/A	0.3617	N/A
	CAM-BERT	3	0.5900	0.5838	0.5280	0.5825
	CAM-BERT	64	**0.6267**	**0.5877**	**0.6002**	**0.5855**
	CAM-BERT	128	0.6251	0.5744	0.5971	0.4194
Kaggle Fake News	BERT	N/A	0.6550	N/A	0.6337	N/A
	CAM-BERT	3	0.8069	0.6408	0.8030	0.6396
	CAM-BERT	64	0.8685	**0.6834**	0.8672	**0.6497**
	CAM-BERT	128	**0.8868**	0.6436	**0.8859**	0.5900
Fake News Net	BERT	N/A	0.7548	N/A	0.4302	N/A
	CAM-BERT	3	0.7557	0.2732	0.4336	0.2400
	CAM-BERT	64	0.7955	0.6468	0.6654	0.6090
	CAM-BERT	128	**0.8028**	**0.7131**	**0.6833**	**0.6302**

Table 6.6: Individual Dataset Experimental Results

non-fake news. The performance gain by using the CAM model compared to BERT is evident since in all of the cases without any model retraining, by only doing fine-tuning, we achieve a performance boost of **25%** on the F1-score.

Analyzing the accuracy and F1 measures in the same dataset and across multiple, we see that these are a bit higher than CS Accuracy and CS F1. One can see that CS Accuracy and CS F1 are dominant in that case. This is because the auxiliary inter-space loss forces the model to learn higher-level concepts and, thus, generalize better for the broader set of problems in cross-domain and cross-dataset manner. Basically, that means that if we fine-tune the model on some dataset, we may later drop the last part and only measure the similarity of the embedding to the trained concept spaces when evaluating it on a different dataset.

(Train) → (Test)	Model	# Dim.	Accuracy	CS Accuracy	F1	CS F1
(Gossip) → (CovidFake)	BERT	N/A	0.5234	N/A	0.3436	N/A
	CAM-BERT	3	0.5234	**0.6145**	0.3436	**0.6052**
	CAM-BERT	64	**0.5375**	0.6064	**0.3806**	0.5874
	CAM-BERT	128	0.5373	0.5691	0.3797	0.4712
(Gossip) → (Politifact)	BERT	N/A	0.5909	N/A	0.3714	N/A
	CAM-BERT	3	0.5909	0.4403	0.3714	0.3699
	CAM-BERT	64	0.6695	0.6259	0.6085	**0.6258**
	CAM-BERT	128	**0.6941**	**0.6468**	**0.6395**	0.6155
(NewsNet) → (CovidFake)	BERT	N/A	0.5234	N/A	0.3436	N/A
	CAM-BERT	3	0.5234	0.6024	0.3436	0.5881
	CAM-BERT	64	**0.5482**	**0.6191**	**0.4064**	**0.6116**
	CAM-BERT	128	0.5480	0.5956	0.4056	0.5354

Table 6.7: Cross-dataset Experimental Results

6.4 Model Interpretability

6.4.1 Explanation of the model decisions using emotions prediction dataset

We hypothesize that if the model really grasps concepts rather than pure contextual relations - we will be able to better classify emotions and extract clusters of emotions related to the sentence by simply doing the kNN on embeddings of the sentence and label definitions.

We have a dataset of 28 emotions. The first step - is to adapt the model to the downstream task - classifying emotions (we only fine-tune the downstream task head here; none of the model weights are updated). We test 4 models: DistilBERT, RoBERTa, CAM-DistilBERT and CAM-RoBERTa. After that, we embed each label definition with each of the models. For the DistilBERT and RoBERTa, we use CLS token representation. For the CAM models, we use Space Mixer representation. This gives us 28 vectors, which we put into the kNN store.

Metric	CAM-BERT	BERT
Accuracy	**0.8309**	0.8220
F1-score (macro)	**0.8006**	0.7484
Precision	0.7126	**0.8876**
Recall	**0.7337**	0.4674

Table 6.8: Social Media Attribution BERT

Next, we compare the results from a classification using the basic fine-tuned model and

using kNN of embeddings. Since the task is multi-label classification, we take the top 1, 3, and 5 nearest neighbors, and the best results happen to fall for the top 3 predictions. However, even for the top 1, results are already better than those for the basic model only.

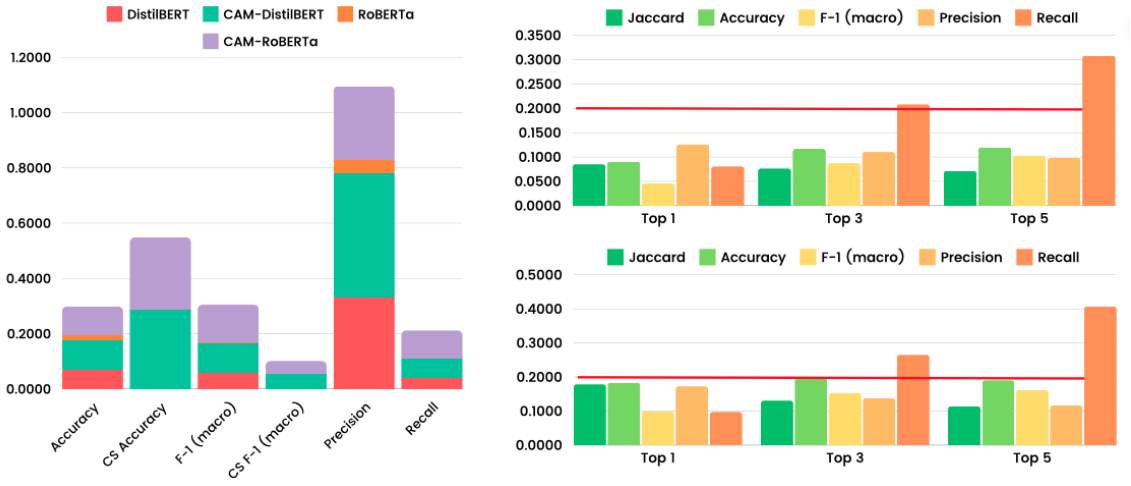

Figure 6.2: GoEmotions Dataset Interpreting the results

We show that with kNN embeddings, we achieve at least twice the boost in Accuracy and F1 for the CAM framework. This proves our hypothesis about better concept capturing and conceptual representation of the sentence. This also helps us explain the decisions made by the model by simply analyzing definitions of the labels 6.10.

Model	Accuracy	CS Accuracy	F1	CS F1	Precision	Recall
DistilBERT	0.0707	N/A	0.0598	N/A	0.3291	0.0388
CAM-DistilBERT	**0.1054**	**0.2868**	0.1065	**0.0553**	**0.4527**	0.0717
RoBERTa	0.0169	N/A	0.0014	N/A	0.0479	0.0007
CAM-RoBERTa	0.1051	0.2614	**0.1370**	0.0457	0.2658	**0.1002**

Table 6.9: Performance Metrics of Different Models on GoEmotions Dataset

Model	# Latent Dims.	Accuracy	F1 Score	Precision	Recall
DistilBERT	N/A	0.1200	0.1025	0.0982	0.3084
CAM-DistilBERT	3	**0.1984**	**0.1623**	**0.1725**	**0.4069**
RoBERTa	N/A	0.1117	0.0627	0.0441	0.1835
CAM-RoBERTa	128	0.1016	0.0599	0.0443	0.1789

Table 6.10: Performance Metrics of Different Models on GoEmotions Dataset 3NN

6.4.2 Social Media Attribution

We compare our CAM Framework with the Social Media Attribution and observe 5.2% F1-score improvement on our own reproducing experiment and almost 2% F1-score improvement compared to the best-reported score from the original paper. The best result there was obtained on the adapted Indian BERT, while we used base uncased BERT without any adaptation, so this performance boost is not exhaustive.

6.5 Context Attribution Layer

As a logical development of our idea that the final dense layer can be replaced with a CAM layer, we arrived at the idea of replacing all of the dense layers in the transformer model with CAM Layers. The assumption states that this would lead to improved overall performance and generalizability. However, after testing this approach (we called it CAM NanoBERT), we arrived at results that were statistically indifferent from the basic BERT results. The explanation for this could be that our framework works best in conjunction with the novel loss. While this heavily relies on the pre-trained LLM embeddings. So, if the embeddings are modified, loss behavior becomes undefined. This particular case is still worth exploring. However, this goes outside the scope of this research.

6.6 Context Attribution with Adapters

A reasonable reference would be to compare CAM and Adapters such as Low-Rank Adaptation (LoRA) and QLoRA. What is the current state-of-the-art training of LLMs? However, we emphasize that those are not mutually exclusive approaches. On the contrary, we encourage exploring how well our external framework combines with LoRA. From the experiments that we conducted, it is reasonable to conclude that CAM, just as for the base model - improves the classification accuracy for LoRA adapters. This agrees with the assumptions we made previously. The better embeddings we can get from the base model, the better projections are found, thus resulting in better classification accuracy. Nonetheless, we want to see how

well these two methods work independently. To do that analysis, we chose the Gemma 2 2B model. We are comparing 20 Million parameters LoRA adapter and 1 Million parameters CAM layer across 3 different datasets: IMDB, Hate Speech, Fake News 6.11. Performance varies depending on the model.

Model	Dataset	Accuracy	F1	Precision	Recall
Gemma 2B	IMDb	0.9679	0.9680	0.9680	0.9679
CAM Gemma 2B	IMDb	0.8003	0.8000	0.8020	0.8003
Gemma 2B	hatexplain	0.7615	0.3198	0.3055	0.3397
CAM Gemma 2B	hatexplain	0.8260	0.4524	0.4130	0.5000
Gemma 2B	fake	0.9975	0.9975	0.9975	0.9976
CAM Gemma 2B	fake	0.9649	0.9648	0.9662	0.9657

Table 6.11: Performance comparison of LoRA Adapter vs. CAM Framework

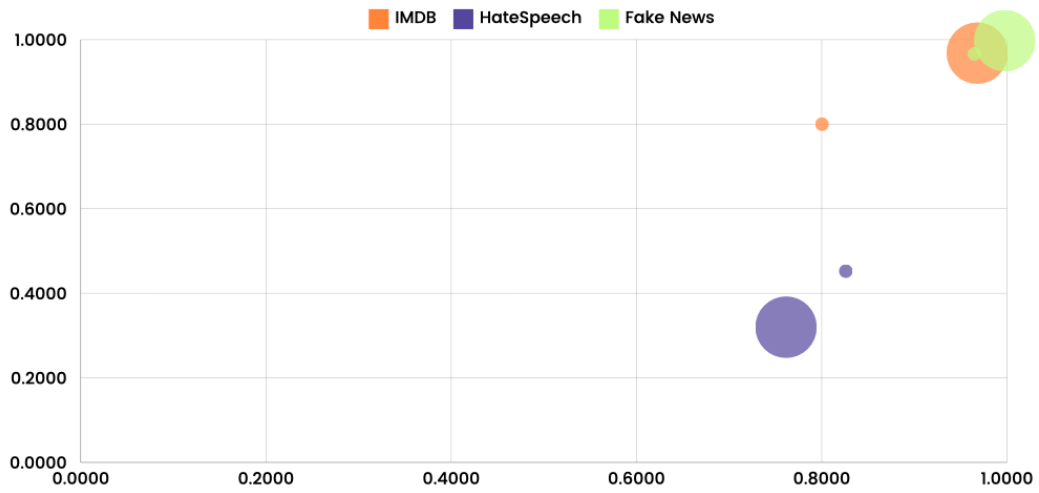

Figure 6.3: CAM-Framework vs. LoRA Adapters (Gemma 2 2B). X-axis - accuracy, Y-axis F1-macro, Size - Model size (1M or 20M)

6.7 Regularization effect on fine-tuning

During the experiments, we observed that the CAM has a much better ratio of recall/precision, which means that it handles imbalanced data much more efficiently. Another observation is that adding just CAM stabilizes the results during training, not allowing the performance to vary a lot between iterations. Find the visualization of the ablation study in 6.5. Additionally, intra-space loss adds more regularization and stabilization and ensures that the concepts in the context attribution will not converge to a single vector.

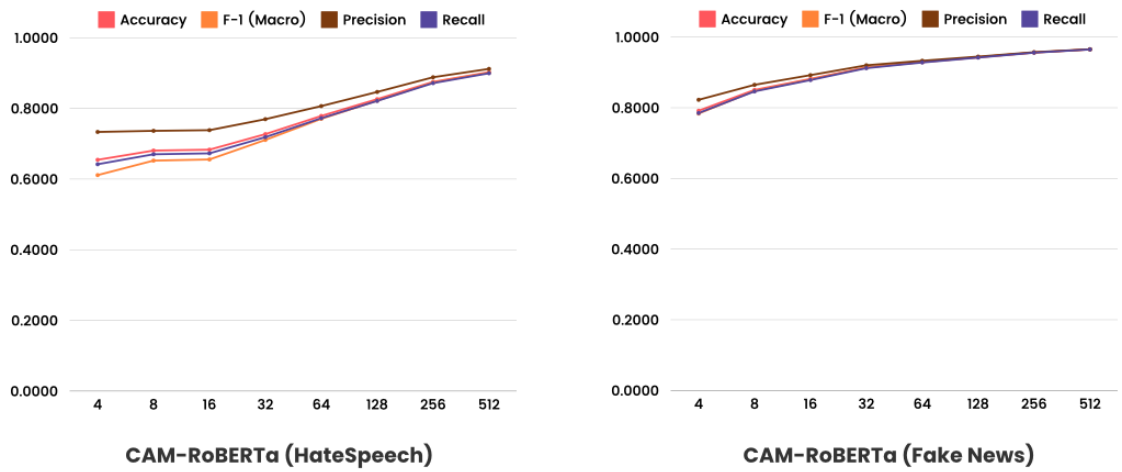

Figure 6.4: Scaling of the metrics compared to the scaling of the latent dimensions

6.8 Ablation study of the Inter-Space and Intra-space losses

To understand the impact of auxiliary losses on model performance and stability, we conducted a comprehensive ablation study. We evaluated two key aspects:

1. The effect of combining auxiliary losses with Cross-Entropy Loss 2. The performance when training exclusively with Inter-Space and Intra-Space losses

Our experiments demonstrate that these auxiliary losses not only enhance the CAM-Model's performance but also provide crucial benefits:

- Improved model stability during training
- Enhanced generalization capabilities
- Better handling of imbalanced datasets

6.8.1 Combined Loss Configuration

In the first configuration, we define the total loss as a weighted sum:

$$L_{total} = L_{CE} + \alpha L_{inter} + \beta L_{intra} \tag{6.1}$$

where α and β are weight parameters, typically set to 0.1 and 1e-5 respectively based on our empirical studies.

Data	# Dims.	Accuracy	CS Accuracy	F1	CS F1	Precision	Recall
GossipCop	3	**0.7566**	0.7053	0.4307	0.6387	0.3783	0.5000
	64	0.2818	0.6857	0.2538	0.6366	**0.5683**	**0.5175**
	128	0.7448	**0.7442**	**0.4436**	**0.6522**	0.4861	0.4985
Fake News Net	3	0.7548	0.6196	0.4302	0.5962	0.3774	0.5000
	64	0.7815	0.7055	0.6016	0.6325	0.7203	0.5928
	128	**0.7973**	**0.7321**	**0.6703**	**0.6370**	**0.7350**	**0.6500**

Table 6.12: Performance evaluation of models trained with combined auxiliary losses on same-dataset scenarios

6.8.2 Exclusive Auxiliary Loss Training

In the second configuration, we trained models using only the auxiliary losses to understand their independent contribution:

(Train) → (Test)	# Dims.	Loss	Accuracy	CS Accuracy	F1	CS F1
(Gossip) → (CovidFake)	3	0.7034	0.5234	0.5292	0.3436	0.3606
	64	0.6892	0.4759	**0.5967**	0.3238	**0.5549**
	128	0.7052	0.4401	0.5543	0.3472	0.4312
(NewsNet) → (CovidFake)	3	0.7046	0.5234	0.5379	0.3436	0.3830
	64	0.6888	0.4761	**0.6260**	0.3238	**0.6039**
	128	0.7055	0.4338	0.5778	0.3467	0.4927

Table 6.13: Zero-shot cross-dataset performance using only auxiliary losses for training

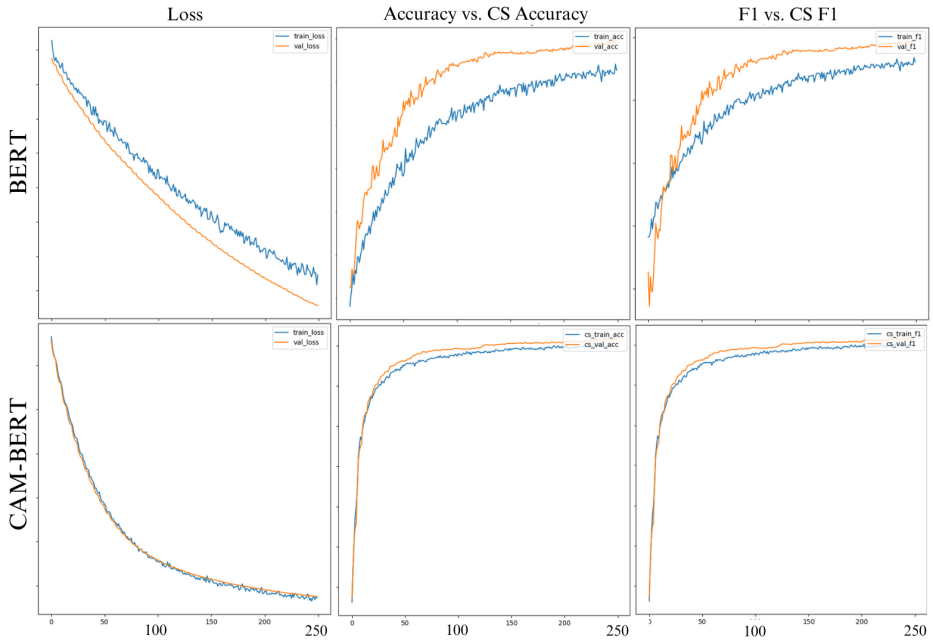

Figure 6.5: Training stability comparison between BERT and CAM-BERT with auxiliary losses

Our findings reveal several key insights:

1. **Training Efficiency**: The auxiliary losses often converge in 10-20% of the epochs required for Cross-Entropy loss convergence, suggesting more efficient training.

2. **Dimensional Impact**: Models with 64 and 128 dimensions consistently show better performance in cross-dataset scenarios, particularly for CS Accuracy and CS F1 metrics.

3. **Loss Collinearity**: In some cases, the entropy objective and auxiliary losses exhibit collinearity, leading to simultaneous improvements in both standard and CS metrics.

4. **Zero-shot Performance**: Training with only auxiliary losses can achieve competitive results in zero-shot scenarios, demonstrating the framework's ability to capture generalizable features without task-specific supervision.

These results suggest that auxiliary losses play a crucial role in both model performance and generalization. The improved stability and cross-dataset performance indicate that these losses help the model learn more robust and transferable features, which is particularly beneficial for zero-shot scenarios and handling imbalanced datasets.

Chapter 7

Results and Discussion

7.1 Discussion

This research focused on the novel methodology towards conceptual embedding for classification with Language models. As an outcome of this work, we have conducted a set of experiments to empirically prove the efficiency of the proposed technique. We have also created the implementation of the proposed framework via PyTorch and provided an open-source GitHub repository to facilitate and simplify future collaboration and exploration. We believe that the potential of this approach is yet to be discovered, and the goal of this paper was to provide some baseline ideas and understanding.

We anticipate improvements by adding more complex transformations after the conceptual projection phase. We also believe that this technique should in no way be limited to classification problems only. The formulation of the regression problem is quite straightforward but needs to be additionally researched. With that, we also expect that 1-to-1 correspondence of the context attribution to the target class is an artificial limitation that we hold in this paper for the simplicity of interpretation. However, if domain knowledge suggests that having multiple context attributions (more than the number of classes) for the task makes sense - then this should also be an option. We would also like to explore further the potential of the interpretation capabilities of the framework and how we can use it to extract knowledge from the model.

7.2 Limitations

While the proposed Context Attribution Model (CAM) showcases significant improvements across various datasets, as evidenced by our experimental results, it is imperative to acknowledge certain limitations inherent to this study and the model itself. These limitations, outlined below, offer avenues for further research and refinements of the approach:

1. **Language Dependence**: Our experiments were conducted exclusively on datasets in English. This limitation raises questions about CAM's performance in processing texts from languages with different syntactic and semantic structures, especially those with low-resource availability or significantly different grammatical constructs compared to English.

2. **Interpretability**: While CAM introduces a geometric interpretation of model predictions, the overall interpretability of how the model attributes context to make these predictions could be further improved. Enhancing the interpretability would not only make the model more transparent but also increase trust in its predictions, especially in sensitive applications.

3. **Data Quality and Bias**: The performance of CAM is dependent on the quality and representativeness of the datasets used for training. Inherent biases in these datasets can affect model performance and fairness. Addressing potential biases and ensuring the diversity of training data is crucial for the development of more equitable and generalized models.

Moving forward, addressing these limitations will be crucial in enhancing the robustness, applicability, and fairness of CAM. Future work will focus on exploring the applicability of the model across diverse domains, languages, and more challenging datasets, as well as efforts to improve model interpretability and reduce computational demands.

Chapter 8

Conclusion

In summary, this study concentrated on a new approach to concept embedding for classification using language models. The research involved conducting experiments to validate the effectiveness of the suggested technique in multiple scenarios with the same dataset and cross-dataset.

We have also introduced a novel loss function that proves to be efficient for the given task and allows better *generalizability*, faster convergence, and more stable and reliable training. This loss function and the model architecture led to the definition of Concept Space metrics, such as CS Accuracy and CS F1. We explain how to evaluate those and how they help us better understand and utilize the CAM Framework. We have also proposed a geometric interpretation of Context Attribution and have proven its efficiency.

Additionally, we implemented the proposed framework using PyTorch and encouraged and streamlined future collaboration and exploration. We are optimistic that the full potential of this method remains to be explored, and the purpose of this paper was to lay the groundwork for ideas and comprehension.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[2] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. 2021.

[3] Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. In Isabelle Augenstein, Spandana Gella, Sebastian Ruder, Katharina Kann, Burcu Can, Johannes Welbl, Alexis Conneau, Xiang Ren, and Marek Rei, editors, *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy, August 2019. Association for Computational Linguistics.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[7] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[10] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986.

[12] Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. Contextual lstm (clstm) models for large scale nlp tasks. *ArXiv*, abs/1602.06291, 2016.

[13] Macedo Maia, Juliano Efson Sales, André Freitas, Siegfried Handschuh, and Markus Endres. A comparative study of deep neural network models on multi-label text classification in finance. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 183–190, 2021.

[14] Steve Durairaj Swamy, Anupam Jamatia, and Björn Gambäck. Studying generalisability across abusive language detection datasets. In Mohit Bansal and Aline Villavicencio, editors, *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 940–950, Hong Kong, China, November 2019. Association for Computational Linguistics.

[15] Ciara Blackledge and Amir Atapour-Abarghouei. Transforming fake news: Robust generalisable news classification using transformers. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3960–3968, 2021.

[16] Francois Chollet, Mike Knoop, Bryan Landers, Greg Kamradt, Hansueli Jud, Walter Reade, and Addison Howard. Arc prize 2024. `https://kaggle.com/competitions/arc-prize-2024`, 2024. Kaggle.

[17] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[19] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at? an analysis of bert's attention, 2019.

[20] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence*. AAAI Press, 2021.

[21] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[22] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[23] Daryl J. Bem. Self-perception theory11development of self-perception theory was supported primarily by a grant from the national science foundation (gs 1452) awarded to the author during his tenure at carnegie-mellon university. volume 6 of *Advances in Experimental Social Psychology*, pages 1–62. Academic Press, 1972.

[24] Bernard Spilka, Phillip Shaver, and Lee A. Kirkpatrick. A general attribution theory for the psychology of religion. *Journal for the Scientific Study of Religion*, 24(1):1–20, 1985.

[25] Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. Methods for measuring, updating, and visualizing factual beliefs in language models. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2714–2731, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[26] Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, page 908–913, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[27] Vijil Chenthamarakshan, Prem Melville, Vikas Sindhwani, and Richard Lawrence. Concept labeling: Building text classifiers with minimal supervision. pages 1225–1230, 01 2011.

[28] Rupak Sarkar, Sayantan Mahinder, Hirak Sarkar, and Ashiqur KhudaBukhsh. Social media attributions in the context of water crisis. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1402–1412, Online, November 2020. Association for Computational Linguistics.

[29] Joe Bolt, Bob Coecke, Fabrizio Genovese, Martha Lewis, Dan Marsden, and Robin Piedeleu. *Interacting Conceptual Spaces I: Grammatical Composition of Concepts*, pages 151–181. Springer International Publishing, Cham, 2019.

[30] Gabriel Erion, Joseph D. Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients, 2020.

[31] Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. Tuning language models by proxy, 2024.

[32] Adi Simhi and Shaul Markovitch. Interpreting embedding spaces by conceptualization, 2023.

[33] Parth Patwa, Shivam Sharma, Srinivas PYKL, Vineeth Guptha, Gitanjali Kumari, Md. Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. Fighting an infodemic: COVID-19 fake news dataset. *CoRR*, abs/2011.03327, 2020.

[34] William Lifferth. Fake news, 2018.

[35] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *CoRR*, abs/1809.01286, 2018.

[36] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *CoRR*, abs/1705.00648, 2017.

[37] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. Goemotions: A dataset of fine-grained emotions, 2020.

[38] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[39] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detec-

tion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875, 2021.

[40] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[41] Dilip Sharma and Sonal Garg. Ifnd: a benchmark dataset for fake news detection. *Complex and Intelligent Systems*, 9, 10 2021.

[42] S.F. Fokkinga and P.M.A. Desmet. *Emotion Typology*. Delft University of Technology, Delft, 2022.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Abstract

Stepan Tytarenko

MS, Fordham University

Context Attribution To Attain Generalizability in Non-fine-tuned Transformer: A Framework for Concepts Excretion in Cross dataset Evaluation Settings

Dissertation directed by Dr. Ruhul Amin, Ph.D

Fine-tuning large pre-trained language models (LLMs) on specific datasets is a prevalent strategy for Natural Language Processing (NLP) classification tasks. However, this process often leads to a significant loss in the models' *generalizability*. To address this, we introduce the Context Attribution Model (CAM), a framework that maintains *generalizability* while improving performance on downstream tasks through task-specific context attribution.

In CAM, we utilize a linear transformation of the text representation from any transformer model. This transformation, guided by a task-specific concept operator, projects the text representation onto a latent concept space—a process referred to as context attribution. The concept operator is optimized during the supervised learning stage using novel loss functions. This approach enhances the discriminator function's capacity, yielding superior classification performance across various tasks.

We evaluate CAM on fake news detection tasks using three distinct datasets and compare its performance against a fine-tuned BERT classification layer baseline. CAM demonstrates significant improvements in both the same-dataset and cross-dataset zero-shot evaluations. Notable results include:

- Covid Fake dataset: 15% accuracy and 18% F1 score improvement.

- LIAR dataset: 3.5% accuracy and 6% F1 score improvement.
- Kaggle Fake News dataset: 23% accuracy and 25% F1 score improvement.
- Fake News Net dataset: 5% accuracy and 25% F1 score improvement.
- In a cross-dataset zero-shot scenario, using a model fine-tuned on the Fake News Net dataset to predict Covid Fake data, CAM achieves a 5% accuracy and 25% F1 boost, underscoring its superior generalizability.
- For the non-pretrained BERT (only trained context attribution operator) we observe **8%** improvement in accuracy and **10%** in F1-score on HateXplain data
- For the IMDB with XLNet base model, we observe an improvement of around **1%** after full model adaptation compared to fully trained vanilla XLNet
- CAM with the base model as DistilBERT non-pretrained (only trained context attribution operator) on IMDB dataset, in a zero-shot manner outperforms basic DistilBERT in the same manner (only head fine-tuning) by **7%** on F1-score
- Compared to the results from the Social Attribution paper (with manual supervision), we observe an improvement of **5.2%** with our model without additional supervision
- We also provide a reproducibility of the results by the code and datasets provided in the repository `https://github.com/StepanTita/space-model`

Additionally, we propose a novel loss function accompanied by auxiliary metrics that enhance generalizability, accelerate convergence, and improve training stability while providing insights into dataset domain representativeness. These metrics provide insights into a dataset's domain representativeness by measuring generalization across various topics and datasets. CAM also offers a geometric interpretation of model predictions by embedding concept definitions (labels), offering a deeper understanding of task-specific context attribution.

VITA

Stepan Tytarenko, son of Viktoria Tytarenko and Andrey Tytarenko, was born on April 16, 2000, in Pervomaysk, Ukraine. In 2021, he earned a Bachelor's degree in Computer Science with a concentration in Software Engineering with High Honors from Kharkiv National University of Radio Electronics, Ukraine. Following his graduation, he commenced his professional journey, gaining extensive experience in software engineering and machine learning across multiple organizations.

From 2020 to 2023, Stepan worked as a Software Engineer and Machine Learning Engineer, where he made significant contributions to distributed systems, CRM optimizations, and AI-powered applications.

In 2023, Stepan joined Fordham University to pursue a Master of Science in Computer Science. During his tenure, he collaborated with Dr. Ruhul Amin on advancing natural language processing and machine learning techniques.

Concurrently, Stepan held engineering roles at adMarketplace, where he spearheaded initiatives to enhance search relevance using advanced embedding models and real-time retrieval systems.

Stepan's thesis focuses on the Context Attribution Model (CAM), a novel framework designed to address limitations in transformer-based architectures for classification tasks. By replacing traditional classification layers in transformers such as BERT, DistilBERT, XLNet, and RoBERTa, CAM enhances both model performance and generalizability. The framework leverages the attention mechanism inherent in pre-trained language models to derive concept attributions from token representations, enabling high-accuracy classification while learning mutually exclusive label-specific concepts.