ARISTOTLE UNIVERSITY OF THESSALONIKI

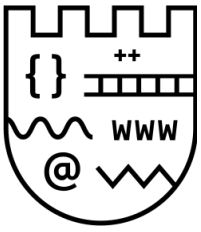
Research paper review through Large Language Models

Αξιολόγηση επιστημονικών δημοσιεύσεων μέσω Μεγάλων Γλωσσικών Μοντέλων

by

Dimitrios Kleitsas - AEM: 3896

Supervisor: Prof. Vlahavas Ioannis



June 2025

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στο πλαίσιο αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Περίληψη

Ο κλάδος της Τεχνητής Νοημοσύνης (ΤΝ), και ειδικότερα η Μηχανική Μάθηση (ΜΜ), έχει εξελιχθεί σε αναπόσπαστο κομμάτι της ζωής του σύγχρονου ανθρώπου, επηρεάζοντας άμεσα την καθημερινότητά του. Από αλγόριθμους που διαμορφώνουν την εμπειρία του καθενός σε διαδικτυακές πλατφόρμες εως και τις προηγμένες εφαρμογές για τη διάγνωση σοβαρών ασθενειών, η τεχνολογία αυτή μπορεί να προσφέρει σημαντικά οφέλη, χωρίς όμως να λείπουν προβληματισμοί για τις επιπτώσεις της.

Ένας από τους σημαντικότερους τομείς της τεχνητής νοημοσύνης είναι η Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ), η οποία αποσκοπεί στην μοντελοποίηση, κατανόηση και παραγωγή γραπτού λόγου μέσω προηγμένων υπολογιστικών μοντέλων. Τα τελευταία χρόνια αυτός ο κλάδος έχει γνωρίσει τεράστια ανάπτυξη λόγω της ανάδειξης των Μεγάλων Γλωσσικών Μοντέλων (ΜΓΜ), τα οποία παρουσιάζουν ικανότητες ερμηνείας και σύνθεσης κειμένων σε επίπεδο που πλησιάζει και, σε ορισμένες περιπτώσεις, ξεπερνάει το ανθρώπινο.

Το σύνολο της επιστημονικής γνώσης αυξάνεται με ραγδαίους ρυθμούς, καθιστώντας δύσκολη την πλήρη κάλυψη, αποτίμηση και αξιοποίησή της. Σε αυτό το πλαίσιο, τα μεγάλα γλωσσικά μοντέλα έχουν την δυνατότητα να δράσουν ως βοηθητικά εργαλεία για την πιο συστηματική και αμερόληπτη αξιολόγηση επιστημονικών δημοσιεύσεων, μειώνοντας σημαντικά τον φόρτο για τους αξιολογητές.

Αυτή η εργασία εστιάζει στην εφαρμογή των γλωσσικών μοντέλων για την επισκόπηση και αξιολόγηση επιστημονικών άρθρων. Στόχος της είναι να εξετάσει το βαθμό στον οποίο τα μεγάλα γλωσσικά μοντέλα μπορούν να υποστηρίξουν σε πρακτικό επίπεδο τη διαδικασία της αξιολόγησης, ως βοηθητικά εργαλεία ή και αυτόνομα συστήματα. Παράλληλα, επιδιώκει να αποτυπώσει μια γενική εικόνα των δυνατοτήτων και των περιορισμών που συνοδεύουν την ενσωμάτωση των μοντέλων στην επιστημονική διεργασία.

Για την αποτίμηση της εφαρμοσιμότητας των μεγάλων γλωσσικών μοντέλων στην αξιολόγηση επιστημονικών δημοσιεύσεων, υλοποιήθηκε μια αρχιτεκτονική δύο σταδίων. Το πρώτο στάδιο αφορά τον αυτόματο διαχωρισμό ενός άρθρου στις κύριες επιμέρους ενότητες του, όπως Εισαγωγή, Μεδοθολογία, Συμπεράσματα, με βάση τα γλωσσικά και δομικά χαρακτηριστικά τους. Αυτό επιτυγχάνεται με την χρήση ενός συνδυαστικού μοντέλου LSTM (Long Short-Term Memory) και BERT (Bidirectional Encoder Representations from Transformers). Τα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου αυτού προέρχονται από τις πλατφόρμες OpenReview και arXiv, οι οποίες παρέχουν πρόσβαση σε πληθώρα επιστημονικών άρθρων που καλύπτουν μεγάλο αριθμό διαφορετικών θεματικών πεδίων.

Στο δεύτερο στάδιο, οι επιμέρους ενότητες και οι πληροφορίες οποίες εξάγωνται από αυτές περνάνε από ένα γλωσσικό μοντέλο το οποίο παράγει την τελική ταξινόμηση του άρθρου ως απόδεκτό ή μη αποδεκτό. Η εκπαίδευση του μοντέλου βασίστηκε σε δεδομένα από την ιστοσελίδα OpenReview η οποία περιλαμβάνει, εκτός από τα ίδια τα άρθρα, την απόφαση αποδοχής ή απόρριψης. Επιλέχθηκαν

δημοσιεύσεις από πλήθος συνεδρίων και θεματολογίων για να διασφαλιστεί η γενικευσιμότητα του μοντέλου.

Με την ολοκλήρωση αυτού του έργου αναπτύχθηκε και αξιολογήθηκε μια ολοκληρωμένη αρχιτεκτονική για την αυτόματη επεξεργασία και αξιολόγηση επιστημονικών δημοσιεύσεων με τη βοήθεια μεγάλων γλωσσικών μοντέλων. Η προτεινόμενη προσέγγιση αναδεικνύει τις δυνατότητες τέτοιων μοντέλων να συμβάλουν στην αποτίμηση βασικών ποιοτικών χαρακτηριστικών, αξιοποιώντας τη δομή και το περιεχόμενο των άρθρων. Τα αποτελέσματα δείχνουν ότι η χρήση γλωσσικών μοντέλων για αυτό τον σκοπό έχει προοπτικές και ανοίγει τον δρόμο για περαιτέρω έρευνα και βελτίωση.

Στο πλαίσιο μελλοντικής μελέτης, προτείνεται η εξέταση μεθόδων για την διερεύνηση της επεξηγησιμότητας των μοντέλων με σκοπό την τεκμηρίωση των αποφάσεών τους, παρέχοντας ουσιαστική ανατροφοδότηση στους χρήστες. Επιπλέον, μια άλλη πιθανή προσθήκη στο σύστημα είναι η μετατροπή των προβλέψεων σε αριθμιτικές βαθμολογίες αντί για δυαδικές τιμές, προσφέροντας έτσι μια πιο λεπτομερή εκτίμηση της ποιότητας του γραπτού.

Αξιοσημείωτος είναι και ο ευρύτερος αντίκτυπος που μπορουν να έχουν τέτοιου είδους συστήματα στην επιστημονική διεργασία και ειδικότερα στην διαδικασία αξιολόγησης. Η ενσωμάτωση προηγμένων γλωσσικών μοντέλων μπορεί να επιταχύνει την ανάλυση επιστημονικού περιεχομένου, να διευκολύνει την πρόσβαση σε πληροφορία και να ενισχύσει την τεκμηρίωση των αξιολογητικών αποφάσεων, ενισχύοντας την αποτελεσματικότητα και την διαφάνεια σε όλα τα στάδια της αξιολόγησης.

Abstract

Artificial intelligence (AI), and in particular Machine Learning (ML), have become an important part of modern society, offering a variety of applications that have had a significant impact on the life of the average person, ranging from the personalization of online content to the diagnosis of life threatening diseases. Among the most impactful areas is Natural Language Processing (NLP), which has enabled the processing, modeling and generation of language through sophisticated autonomous systems. Large Language Models (LLM) represent the latest big breakthrough in the field, offering capabilities of language understanding and production at a higher level than ever before. As scientific output continues to grow at a very quick pace, it becomes increasingly difficult to keep track of all latest developments in a field and assess the new knowledge. Large language models have the potential to aid in the systematic and unbiased review of research publications and serve as a semi-automated tool for certain parts of the peer review process, providing helpful feedback and corrections. This thesis explores the usage of LLMs for the automation of the review process of scientific papers. A two-step architecture is proposed: the first step entails the segmentation of the work into its main sections using a hybrid LSTM and BERT model, the second is tasked with performing section-level classification of the paper as either accepted or rejected. To convert these predictions to the paper level, several aggregation strategies are considered to ensure the quality and accuracy of the final decision of the system. This approach aims to contribute toward the development of more efficient, scalable, and consistent tools for supporting the academic peer review process.

Acknowledgements

I would like to express my sincere gratitude to Professor Ioannis Vlahavas for the opportunity to undertake this thesis. His guidance, structured approach and insightful advice proved instrumental to the successful completion of this work.

I am also very grateful to PhD candidates Georgios Liapis and Eleftherios Kouloubris, members of the Intelligent Systems Lab at the Aristotle University of Thessaloniki, for their mentorship and support during every stage of this thesis. Their consistent feedback and encouragement played a crucial role in shaping the quality of this work.

Finally, I am very thankful to my friends and family for helping me stay sane and focused during this journey. Their support has been vital for the completion of this academic endeavour and I am truly grateful for their presence in my life during this important chapter.

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence	Τεχνητή Νοημοσύνη
ML	Machine Learning	Μηχανική Μάθηση
SL	Supervised Learning	Επιβλεπόμενη μάθηση
RL	Reinforcement Learning	Ενισχυτική μάθηση
MSE	Mean Squared Error	Μέσο Τετραγωνικό Σφάλμα
MAE	Mean Absolute Error	Μέσο Απόλυτο Σφάλμα
SVM	Support Vector Machine	Μηχανή Διανυσμάτων Υποστήριξης
NN	Neural Network	Νευρωνικό Δίκτυο
DL	Deep Learning	Βαθιά Μάθηση
MLP	Multi-Layer Perceptron	Πολυεπίπεδο Αντίληπτρο
CNN	Convolutional Neural Network	Συνελικτικό Νευρωνικό Δίκτυο
RNN	Recurrent Neural Network	Αναδρομικό Νευρωνικό Δίκτυο
LSTM	Long Short-Term Memory	Μακρά Βραχύχρονη Μνήμη
NLP	Natural Language Processing	Επεξεργασία Φυσικής Γλώσσας
GPT	Generative Pre-trained Transformer	Παραγωγικός Προεκπαιδευμένος Μετασχηματιστής
GPU	Graphics Processing Unit	Μονάδα Επεξεργασίας Γραφικών
LLM	Large Language Model	Μεγάλο Γλωσσικό Μοντέλο
PEFT	Parameter-Efficient Fine-Tuning	Αποδοτική Παραμετρική Βελτιστοποίηση
LoRA	Low-Rank Adaptation	Προσαρμογή Χαμηλής Τάξης
RAG	Retrieval-Augmented Generation	Παραγωγή Ενισχυμένη με Ανάκτηση
BERT	Bidirectional Encoder Representations from Transformers	Αναπαραστάσεις Κωδικοποιητή Διπλής Κατεύθυνσης από Μετασχηματιστές

Chapter 1

Introduction

Machine learning (ML) has transformed how computers process and understand human language. Natural language processing, a branch of artificial intelligence, enables machines to interpret and generate text, with modern language models that can now perform complex tasks like translation, summarization, and even creative writing with high accuracy. These advances have opened up possibilities for automating text-based tasks across many domains.

The scientific publishing process faces an exponential increase in the volume of research papers, with the peer review system struggling to keep up. Traditional manual review, while thorough, is slow and can be inconsistent across different reviewers. Given the demonstrable capabilities of large language models in handling complex text, an opportunity is presented to explore whether these models can assist with evaluating scientific articles, potentially helping to streamline the review process without replacing human expertise.

This thesis develops a two-stage system for automatically processing and evaluating scientific papers. The first stage segments articles into their main sections using a hybrid approach that combines LSTM networks with BERT. This segmentation process uses both the language patterns and structural features of academic writing to identify key sections accurately. The second stage employs a fine-tuned BERT model to classify whether an article should be accepted or rejected. This classifier is trained on real data from OpenReview, which contains papers and their actual acceptance decisions from various conferences across different fields. The system is built using PyTorch for deep learning implementation and leverages Hugging Face's transformer library for pre-trained models and dataset management. This setup allows for efficient experimentation and model optimization.

This thesis is comprised of four chapters, each addressing a distinct aspect of the research. Chapter 2 provides an overview of the core methods in Artificial Intelligence and Machine Learning that this work is based on. It introduces key concepts such as Natural Language Processing (NLP) and the transformer model. Chapter 3 focuses on the details research process and explores the role of large language models in scientific knowledge management. It outlines the peer review process and the challenges inherent in evaluating academic publications, before delving into the principles behind LLMs. This chapter also discusses model adaptation techniques such as fine-tuning, quantization, and parameter efficient training, concepts which are critical for the effective deployment of LLMs in domain-specific tasks. Chapter 4 presents the implementation of the proposed system. It details the architecture and development of the

pipeline, including the segmentation of article sections and the subsequent classification using a BERT-based model. The chapter describes the data collection and preprocessing steps, the training setup using PyTorch and Hugging Face tools, and the experimental evaluation based on OpenReview data. Finally, Chapter 5 concludes the thesis with a summary of the findings, a discussion on the strengths and weaknesses of the proposed approach and ideas for future research directions.

Chapter 2

Artificial Intelligence methods

In this chapter, we examine the the algorithms, tools and techniques used in this thesis, as well as its motivations, focusing on the breakthroughs that have rendered the creation of such sophisticated systems possible. In today's digital, information-oriented society, data has become a commodity, labeled as the world's most valuable resource (The Economist, 2017). But raw data does not stand on its own, nor can humans be expected to manually parse millions of bits of information and extract accurate conclusions. This discrepancy between the overabundance of data and the lack of means to efficiently process it has been addressed through Machine Learning (ML) (Khan, 2024). In the following section, some common machine learning paradigms, frameworks, methodologies, and models are examined, providing necessary foundational knowledge.

2.1 Machine Learning

Machine Learning, a subfield of Artificial Intelligence (AI) (Russell and Norvig, 2021), encompasses the sum of algorithms where statistical models are trained on large quantities of data in order to obtain the capacity to generalize on situations that they have not faced before. In contrast to symbolic AI, Machine Learning does not require substantial prior human expertise and direct instructions, only a sufficiently big dataset. ML has had an indisputable impact on countless sectors ranging from academia and industry to society as a whole. Medicine and healthcare, finance and marketing, education, energy, entertainment. The potential applications are vast and the field is still far from reaching its full potential.

Before proceeding, an important caveat in regard to these algorithms must be addressed. ML models posses neither the agency nor the common sense to dissect implicit biases and factual inaccuracies in the information they process. If the dataset they receive is skewed, discriminatory, or full of errors, these properties will manifest in the outputs of our models. This raises multiple ethics concerns regarding the reliability and safety of these systems. Their purpose should be to drive progress and provide solutions that have a positive effect on our society, not to propagate stereotypes and amplify inequity. It is our responsibility as students, researchers and enthusiasts of this technology to ensure that our work remains grounded in these moral principles.

The different methods used to train ML models are commonly separated into the following three types:

- **Supervised Learning (SL)** (Jo, 2023b): The model is trained on a labeled dataset, where each data sample is accompanied by the category it belongs in or a numerical value that represents it.
- **Unsupervised Learning** (Wang, 2025): The model is trained on unlabeled, raw data.
- **Reinforcement Learning (RL)** (Yan, 2023): The model is trained by making decisions in a simulated environment.

These categories are not entirely rigid and they are often used in tandem to procure better results. For instance, in Semi-Supervised Learning, models are trained with both labeled and unlabeled data. A deeper dive into the main training paradigms follows.

2.1.1 Supervised Learning

In a Supervised Learning setting, each sample needs to be paired with a label. The construction of such datasets is a laborious process and cannot always be efficiently automated, rendering them a costly option. The advantage of SL models is the element of ground truth, which the other methods generally lack. It is a very powerful paradigm that has been shown to consistently produce great results, given sufficient data.

Some common examples of labeled data include pictures of animals along with their type, a snippet of a song and its genre, the contents of an email and whether or not it is classified as spam, the value of a stock for the n previous days and its value today. A SL model trained on such data would aim to learn a general function that correlates input samples to their labels and is able to repeat the process on data points that it has not seen before.

Supervised Learning problems are sorted into two subgroups, classification and regression (Bartz-Beielstein, 2024), depending on whether their output is a discrete label or a mathematical value. In these problems, a model, denoted as f, is trained to map data samples with n features, denoted as $\mathbf{x} = [x_1, x_2, ...x_n]$, to their corresponding outputs, denoted as y.

Classification is the process by which observations are separated into distinct categories. Given a set of n classes C $= \{C_1, C_2, ..., C_3\}$, classifications is defined as $f(\mathbf{x}) = y, y \in C$. Classification can be divided into two main types, binary and multi-class, depending on whether there are two or more than two classes. The performance of a classification model can be evaluated as the ratio of correct categorizations to total predictions, which is referred to as accuracy, or with more sophisticated methods such as cross-entropy, which also accounts for the certainty with which the model makes its predictions.

Regression, the second big discipline in Supervised Learning, is the process by which a model estimates the mathematical value of a certain dependent variable based on the values of one or more independent ones. Assuming there are no range restrictions to the output value, regression is defined as $f(\mathbf{x}) = y, y \in R$. Mean Squared Error (MSE) and Mean Absolute Error (MAE) (Jadon et al., 2022) are commonly

used to evaluate the model outputs. MSE squares the difference between the output and the true value, penalizing large errors more. MAE returns the absolute difference, punishing all errors at the same rate.

So far we have discussed ML in terms of models. A model is a mathematical function whose goal is to capture the relationship between input and output. There is a great variety of models, and selecting the correct one for any given problem depends on the nature of the task and the type of data that is available. Some frequently used Supervised Learning models are examined below.

Linear models

Linear models are some of the simpler and more interpretable ML algorithms available, while still performing adequately for a decent variety of tasks. Linear Regression (Ahlawat, 2025) is the process by which a line, represented by $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$ is fit through the data. The points on this line, then, represent the value of the model's output y for any given combination of parameters $(x_1, x_2, \ldots, x_n)$. An example of linear regression can be seen in Figure 2.1.

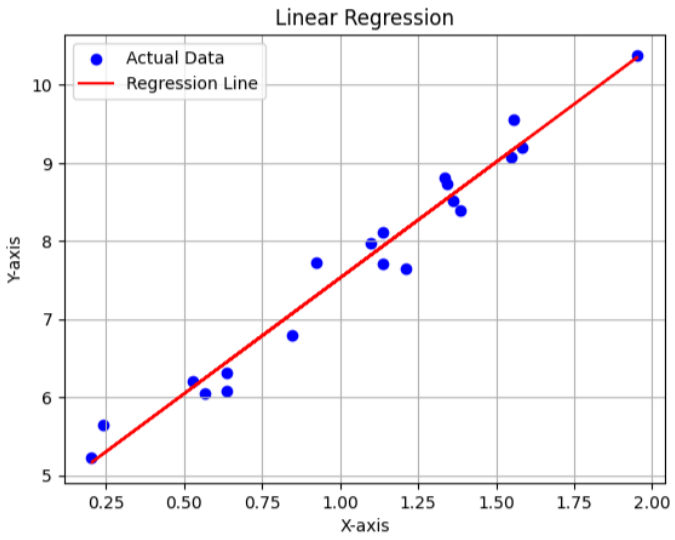


FIGURE 2.1: Linear Regression (OC)

Logistic Regression (Backhaus et al., 2023), similarly, creates a line or, in higher dimensions, a hyperplane, whose purpose is to separate the data into two groups, with points on the opposite sides of it belonging to different classes. It is, despite its name, a classification model. While these models require little data and are fast to prototype and train, their performance suffers when the relationship between input and output is complex and not linear.

Decision trees

Decision Trees (Zollanvari, 2023) are a category of highly interpretable and data efficient ML algorithms that use a tree-like structure that consists of nodes, branches and leaves. The construction of a decision tree begins at the root node, where the dataset is split based on a specific feature. The feature is selected according to a pre-decided criterion, such as maximizing information gain. Proceeding from the root node, the data flows down the branches of the tree and is recursively divided into smaller and smaller subsets, until a predefined depth is reached or until the model can no longer improve from further splits. The final nodes, known as the leaf nodes, represent the categorization of the data samples that reach them. An example of a decision tree can be seen in Figure 2.2.

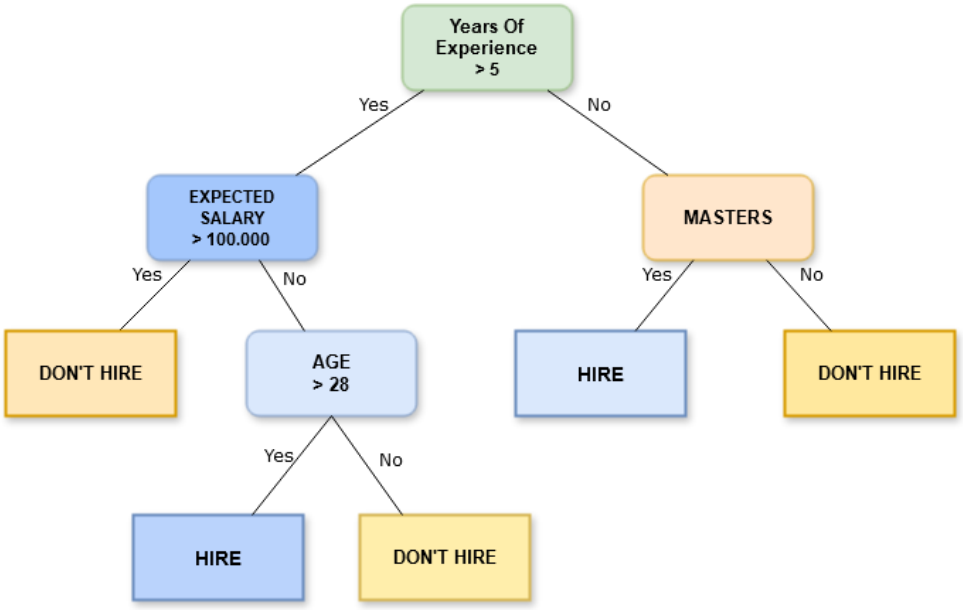


FIGURE 2.2: Decision Tree (OC)

Support Vector Machines (SVM)

Support Vector Machines (Hearst et al., 1998) exist in both linear and non-linear form. A linear SVM attempts to split the plane into two, with either side representing a separate class, by maximizing the margin between the data points of each category. Cases where the data is linearly inseparable are addressed by non-linear SVMs and their utilization of the kernel trick. This trick relies on the assumption that a feature space that is not linearly separable in the current dimension, would be so in a higher one. To explicitly transform the data to a higher dimension, however, is incredibly computationally expensive. With the kernel trick, only the computation of the dot product of two features in that higher space is required. Popular kernels include the Polynomial, Radial Basis Function, and Sigmoid.

Neural Networks (NN)

The importance of Neural Networks (Joshi, 2023) to modern Machine Learning cannot be overstated. They are the cornerstone of many recent developments in the field, and many endeavors, including this thesis, would simply not be possible without them. Deep Learning (DL) (LeCun et al., 2015) is the term used to describe the subcategory of ML where NNs are used to discern complex patterns in data.

Neural networks consist of multiple layers of artificial neurons connected by edges, inspired loosely by the neurons and synapses in the human brain. These models, although more computationally expensive and difficult to train compared to simpler ML models, have emerged as a leading method due to their versatility, robustness and ability to deal with increasingly complex problems.

One of the earliest forms of NNs is the Multi-Layer Perceptron (MLP) (Olivieri, 2024). The MLP consists of layers of fully connected neurons with non-linear activation functions in between. Each neuron contains a weight w and optionally a bias b for every neuron in the next layer, and the output for an input sample x will be $y = \sigma(wx + b)$, where σ is the chosen activation function.

An activation function is a mathematical operation that transforms a given input value. They are used in neural networks to add an element of non-linearity, allowing them to solve more complex problems. Some common examples of activation functions in both mathematical and graphical formulation can be seen in Table 2.1.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.1) \qquad \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (2.2) \qquad f(x) = max(0, x) \quad (2.3)$$

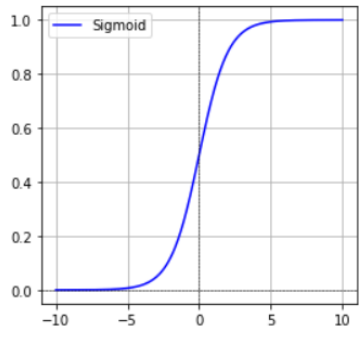


FIGURE 2.3: Sigmoid activation (OC)

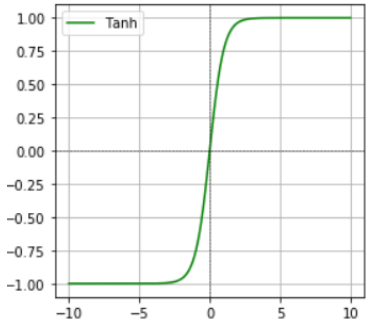


FIGURE 2.4: Hyperbolic Tangent activation (OC)

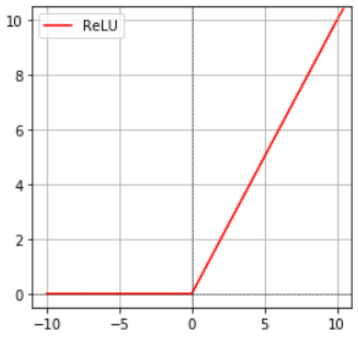


FIGURE 2.5: ReLU activation (OC)

TABLE 2.1: Sigmoid, Hyperbolic Tangent, ReLU activation functions

The Softmax activation function represents a special case, as it is typically only used at the final layer of a network that performs multi-class classification and serves to convert the raw numerical values, also known as logits, into probabilities that represent the likelihood with which the model believes a sample belongs in any given category.

Neural networks are trained using two fundamental concepts: backpropagation and gradient descent (Alake, 2022). A forward pass through the network is completed and for any given set of inputs the model produces a corresponding set of outputs. Comparing these against the labels, a measure of how far the model's predictions are from the ground truth is extracted. This is quantified by a single value, referred to as the *loss*. The process continues by going backwards through the layers of the network, calculating the derivative of each neuron at every layer in order to find out which ones affected the result the most. Using gradient descent, the weights are altered by a small amount in a direction that would decrease the loss of the model. This same process is followed over our entire dataset for a set number of epochs or until a stopping condition is met.

Following the creation of the Multi Layer Perceptron, the Convolutional Neural Network (CNN) (Pattanayak, 2023) and the Recurrent Neural Network (RNN) (Jo, 2023a) were the next noteworthy advancements in deep learning architectures. MLPs, despite representing a great breakthrough, struggle with more complicated types of structured or sequential data and have limited scalability. The new architectures were each created to address shortcomings of the original model and provide better solutions to more specialized problems.

The primary strength of CNNs is image and video processing, as they are designed to deal with data that is structured in a grid, like pictures. By parsing a filter over the grid-like data and applying a convolution operation, they capture not only the values of each cell but also the spatial relationships between them. Additionally, rather than assigning an individual neuron to every value of a sample, only a single neuron is required for each square in the filter.

The MLP and CNN are both categorized as Feed Forward Neural Networks. This means that they follow a straightforward and predefined path from input to output and after a single pass a result is produced. Recurrent Neural Networks have a completely different architecture, employing a loop-like structure, which allows them to better deal with sequential data. Given, for instance, a sentence or a time series, the RNN will dissect it into parts and process them one by one. At each following step, the network collects information from the the previous iteration and stores it into the *hidden state*, which is added to the inputs of the next run-through. This allows it to extract information from past inputs. A simple RNN can be examined in Figure 2.6.

One of the core limitations of the early RNN models was their inability to retain long-term dependencies, due to a problem known as vanishing/exploding gradients (Bengio et al., 1993). The updates applied to the weights of the network depend on the multiplication of multiple gradients. As the network processes more time steps, the more successive multiplications are done between the gradients, resulting in either tiny changes which are not enough to influence the training in a noticeable way, or extremely large ones that completely offset the model.

Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks were created to address this challenge. As the name suggests, LSTMs feature a short term memory similar to that of the

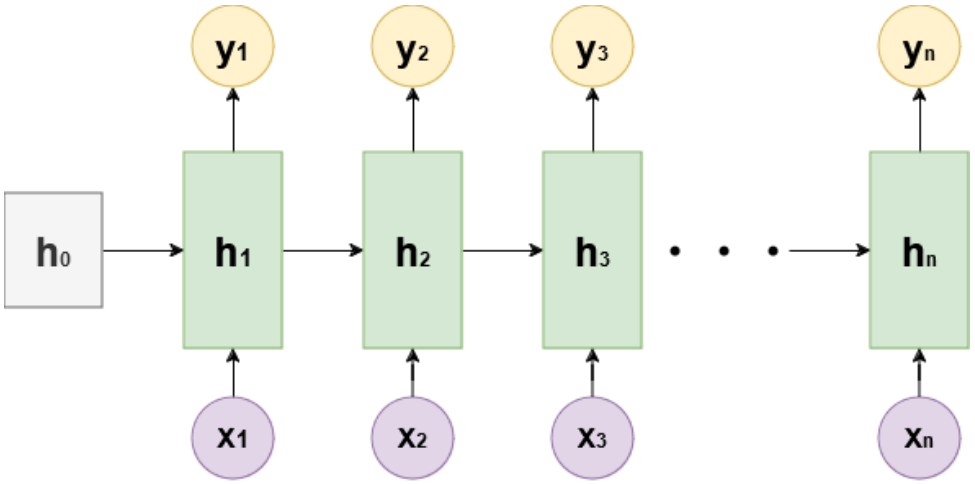


FIGURE 2.6: Unfolded Recurrent Neural Network (OC)

RNN but also a long term memory which persists for longer. The main improvements of the LSTM over the RNN are twofold: First, the long term memory is not modified directly by any weights or biases and thus is not subject to vanishing/exploding gradients. Second, the LSTM has mechanisms called gates that can decide how much of the knowledge they have acquired to store and how much to forget, allowing it to hold on for longer to information it considers important and forget outliers in the data more easily.

While LSTMs address the issue of vanishing/exploding gradients, they are not immune to them. Additionally, due to their sequential structure, their training cannot be parallelized efficiently. The Transformer (Vaswani et al., 2023) architecture has emerged as a solution to these issues and has enabled the creation of much larger scale systems. At a later stage an entire subsection is dedicated to the transformer, as it is at the core of this thesis.

2.1.2 Unsupervised Learning

Unsupervised Learning (Wang, 2025) uses data that is not clearly labeled and can be gathered in large quantities with relative ease, eliminating the need for manual annotation. Here, the purpose of training is not to learn the connection between input and output, but to detect patterns or groupings in the data. Although this approach provides increased flexibility, these models operate without ground truth labels to validate their findings and are used for an entirely different category of problems compared to SL.

Clustering (Runkler, 2025) is the process of splitting data into groups, which are referred to as clusters. Data points belonging in one cluster will have more in common between them on certain attributes than with those in other clusters. The basis for this separation is dependent on the requirements of the problem. This differs from classification in Supervised learning, as knowing beforehand in which class a given sample belongs is not required. On the contrary, clustering might aid in the identification of categories and patterns in the data that were previously unknown.

Dimensionality Reduction (Sarang, 2023), another common use of Unsupervised Learning, aims to reduce the size of a piece of data while maintaining its core properties. Image compression, for example, falls under this discipline and while the traditional solutions like JPEG and PNG compression do not utilize ML, neural network based approaches are gaining traction in the research community, with the Autoencoder (Bank et al., 2021) being one such example.

Autoencoders are an neural network architecture that features an encoding and a decoding function. In the encoder, the data is compressed down into its principal components. Their aim is to learn an efficient representation of a sample that can subsequently be taken by the decoder and reconstructed back into the original. Their use, however, is not limited to data compression. The encoded version contains all the important elements of the data point while requiring significantly less space and computational power. By running an entire dataset through an Autoencoder, a the minimal representation of every sample is obtained, and this new set can be used to train other ML models. This can speed up the training significantly and boost performance as the model will have to deal with less irrelevant information and noise. A toy example of the effect of an Autoencoder on a data sample is visualized in Figure 2.7.

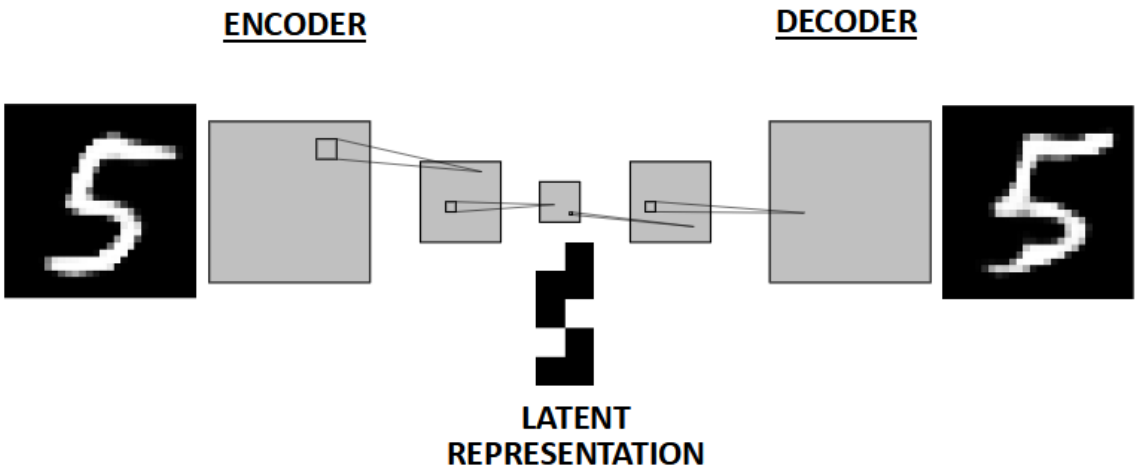


FIGURE 2.7: Autoencoder (LeNail, 2019)

The Variational Autoencoder (Kingma and Welling, 2019), a modification on the standard Autoencoder, seeks to create a mapping of the dataset to a distribution, rather than from point to point. They learn the latent space the data belongs in, and then sample from that to recreate the original. The advantage is that once the model has learned the latent representation, it can synthesize entirely new data similar to its inputs.

2.1.3 Reinforcement Learning

Reinforcement Learning (Yan, 2023), the third and last training framework, does not require explicit data, but only a simulation of the process that the model must learn. Conventionally, in RL models are referred to as agents and the simulation as the environment. The training proceeds as follows: The agent observes its environment and selects an action to perform based on its policy. This decision affects the

environment and provides the agent with a reward or a punishment, incentivizing the model to choose the best possible actions under the circumstances it is presented with. This behaviour is categorized as a Markov Decision Process (Xiao, 2024) and is how RL problems are typically modeled. An agent is trying to maximize its rewards in an environment where the future states depend on the current state and on the agent's actions.

Neural Networks have emerged as the predominant architecture for RL models. Their capabilities as universal function approximators become increasingly important as state and action spaces grow too large to model exhaustively. While Deep RL has made significant advancements in research and has been used effectively under specific circumstances, there are persistent questions regarding its safety and reliability that prevent it from achieving large scale deployment and wide use. In practice, rather than a model being trained from scratch with RL, it is often first trained with the more traditional supervised or unsupervised learning approaches and RL is used as a supplementary process to attune the model more to the requirements.

2.2 Natural Language Processing

Natural Language Processing (NLP) (Igual and Seguí, 2024) is a subfield of AI concerned with analyzing, interpreting and generating human language. Language is at the epicenter of all human interaction. It is how communication is conducted, ideas are conveyed and knowledge is spread. Instilling an understanding of language into AI is a topic that has interested computer scientists and researchers for a very long time. The transition from symbolic, rule based models towards statistical and later neural ones has generated a massive boom in the efficiency, but also the complexity of these systems. Tasks typically associated with NLP are sentiment analysis, text summarization and generation and many more.

2.2.1 Natural Language Processing problems

Sentiment Analysis (Goniwada, 2023) is the process of determining the emotional tone that is expressed in a piece of text. The goal is to ascribe a certain emotion or opinion to a sample, such as positive, neutral or negative. More advanced models can provide fine-grained labels and identify more specific emotions. Traditional methods relied on predifined rules and lists of words, while more modern methods use deep learning models that can understand the context and nuances of the interactions.

Text Summarization (Jo, 2024) is the process by which a language model summarizes a large document into a smaller text while maintaining the key information from the original. This can be done with extractive methods, by selecting entire sentences of phrases from the original to keep, or abstractive methods, by paraphrasing the document and generating new sentences that convey the original meaning.

Machine Translation (Qamar and Raza, 2024) is the task of converting text from one language into another. Early models offered exact word for word translations and tended to miss a lot of the necessary

context and meaning. The advancement of statistical and deep learning NLP models trained on extensive datasets significantly improved the quality of translations by being able to capture dependencies between words and account for multiple meanings.

Chatbots and Question Answering Systems (Singh et al., 2021) seek to provide answers to questions posed by users in natural language. Traditionally, this would be achieved by examining a large corpus and extracting passages directly from them. Nowadays, the more advanced neural-based systems generate the answers themselves, as much of the knowledge required to answer questions is embedded into their weights.

Part-of-Speech Tagging (Pandey, 2023) is another notable NLP task and it involves labeling the words in a sentence with their grammatical roles, such as noun, verb or adjective. Named Entity Recognition (Pandey, 2023) has a similar function and strives to categorize the different entities in a text. Entities can include people, organizations, locations or any other predefined category.

Regular Expressions (Friedl, 2006), while not inherently a part of Natural Language Processing, are often used alongside NLP to assist with preprocessing. They consist of a sequence of characters that define patterns used for searching, matching and editing text and are often applied for tokenization, cleaning data or extracting specific information. For instance, the expression [a-zA-Z0-9\s] can be used to locate all alphanumeric and whitespace characters in an excerpt of text.

2.2.2 The evolution of NLP models

The earliest NLP models relied entirely on huge sets of hand-written rules. Although they represent the logical first step into language processing, they are fundamentally limited and cannot account for the sheer complexity and size of language, where even minor changes in the order of words can completely alter the meaning of a sentence. Handling every slight variation, exception or grammatical error would require an impossible amount of rules. These early systems could manage simple interactions with predetermined responses but could not be scaled to comprehend authentic use of language.

Statistical models (Johnson, 2009), the next big evolution for NLP, rather than relying on hand-crafted crafted specifications, treat language as a probabilistic system and learn from large amounts of data. This shift was aided by the rapid advancement of computational resources that enabled the construction of operationally demanding systems, as well as the meteoric rise of the web, which provided easy access to enough written and spoken language to train incredibly large models.

Towards the end of the life cycle of Statistical NLP, the best performers within the paradigm were the n-gram class of models. As the name suggests, they rely on the n previous words in the sentence in order to generate the most probable follow up based on the frequency with which the words were found together in the dataset the model was trained on. Statistical NLP, while a huge improvement from its symbolic counterpart, still faced significant limitations. Processing or generating entire paragraphs of

text coherently was still beyond reach and many of the intricacies and nuances of language could not be captured.

MLPs were the first neural network models applied to NLP, but it was RNNs that kickstarted the neural era of natural language processing (Mikolov et al., 2010). Designed specifically to handle sequential data, they immediately began to outperform their statistical counterparts. LSTMs followed, providing noticeable improvements in capturing long term dependencies but without solving the major issue present in these sequential models: the inability to parallelize training. The architecture that does not suffer from this issue and has radically revolutionized NLP is the transformer, and in the next section we will explain why.

2.3 Transformers and the Attention mechanism

The Transformer (Vaswani et al., 2023) neural network architecture was initially designed for the task of machine translation, where it outperformed many of the then state-of-the-art models. In the time since, transformers have fundamentally transformed the field of natural language processing, setting new benchmarks across numerous language tasks and have successfully been adapted to many other domains.

2.3.1 The standard Transformer

The transformer initially splits the received text into distinct tokens. These tokens may be full words, syllables, symbols or numbers, depending on the vocabulary the model possesses. As neural networks cannot process raw characters, these tokens are converted into vectors via a process known as word embedding. Each token has multiple embeddings in order to account for the different meanings it may hold. For instance, a ”bat” in a cave and a ”bat” in a baseball player’s hands mean completely different things, but they would be indistinguishable to a model without the above step. The embeddings are learned by shallow feed-forward neural networks during the training phase.

Equally important to the meaning of each individual word, is its order in the sentence. The transformer must be able to reliably distinguish between two reordered instances such as ”John cooked food” and ”Food cooked John”. This is where positional encoding becomes essential. It works by adding a numerical value to the word embeddings, altering their meaning to reflect their position in the sentence. While the positional encodings can also be learned, in the original transformer paper they were done by representing every encoding dimension with a sinusoidal wave of a different frequency and sampling from it.

Once embedded and encoded, the input flows into the encoder of the transformer and enters the first self-attention block, which consists of multiple attention heads. In an attention head, the model examines the other tokens around a certain word and extracts semantic information from them to shift its meaning. This can be seen in Figure 2.8. The process is repeated for all the available words. The

attention-affected values then enter a fully connected feed forward neural network. The encoder is comprised of many such pairs of attention blocks and fully connected networks. Additionally, a residual connection is established between the position encoded values and the attention values, and the attention values and the feed forward network's output. Both sections also feature a layer normalization step.

FIGURE 2.8: Attention between tokens (Vig, 2019)

The decoder functions similarly, except for a few important differences. First, rather than standard attention heads, it has masked attention heads. This means that a word that comes after another in the sequential order cannot affect the meaning of the previous word. Second, attention values are added from the encoder to connect the meanings of the words. This is referred to as cross attention or encoder-decoder attention, as it bridges the two segments. Upon exiting the decoder, the values undergo a linear transformation and are then put through a softmax function, to represent the output as a probability distribution over the possible tokens. A helpful visualization of the entire transformer architecture can be seen in Figure 2.9.

2.3.2 Generative Pre-Trained Transformers

The Generative Pre-trained Transformer (GPT) (Yenduri et al., 2023), a decoder-only variation of the original transformer model, achieved rapid success and facilitated the creation of numerous tools that have altered countless sectors of industry and academia. *Generative* refers to its capability to procedurally generate new text and *Pre-trained* alludes to the fact that the model has been trained on massive quantities of data and has the option to be further adjusted to more specific tasks.

The GPT's general function is the prediction of the next word in a sequence, with its input being is all the words that precede it. While this may appear limiting, the abstraction of generating whole, coherent sentences into simply predicting a single token has proven to be extremely effective. As the GPT does not have an encoder, every previous token up to a set context size is processed in the Decoder, using masked attention heads on both the input and the output and without a cross-attention step.

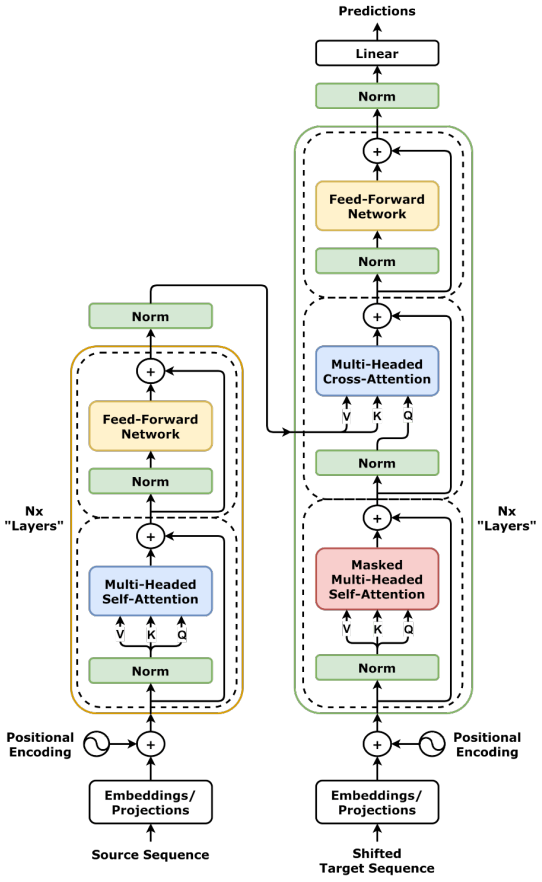


FIGURE 2.9: Transformer architecture illustration (dvgodoy, 2021)

An important thing to note is that all steps in both the GPT and the standard Transformer do not have to follow a sequential process. The values required are initially available, which means that all tokens can be processed in tandem. Utilizing modern Graphics Processing Units (GPUs), the training of these models can be parallelized and sped up significantly, giving them a huge edge over older, recurrent architectures. The advances in research and scaling of these systems have facilitated the creation of incredibly powerful models with billions of parameters that have redefined the state-of-the-art in NLP. These models have completely transformed AI in its entirety, and we will now examine how.

2.4 Large Language Models

Large Language Models (LLM) (Naveed et al., 2024) represent a big step forward in natural language processing and are characterized by their ability to understand and generate human-like text. Built upon the foundation of transformers, LLMs are pre-trained on vast datasets in order to learn intricate patterns and relationships in language. This allows them to generalize across a wide range of tasks with minimal or no task-specific training. By utilizing massive computational resources and billions of parameters, LLMs have completely radicalized the approach to language-related challenges.

To understand the rapid pace with which LLM's have progressed in the past few years, OpenAI's GPT-n series is examined:

- **GPT-1** (Radford et al., 2018): GPT-1, featuring 117 million parameters and trained on 4.5 gigabytes of raw data, was one of the largest and most powerful language models at the time. While it introduced many key concepts, like the paradigm of training on large quantities of data and then further fine-tuning on more specific tasks, its struggle to generalize and very limited scope indicated that the technology was still in its early stages.
- **GPT-2** (Solaiman et al., 2019): GPT-2, with 1.5 billion parameters and trained on 40 gigabytes of text, was a marked improvement over its predecessor. As the potential of such technologies grew more evident, ethical concerns were raised due to the harmful content that could be mass-generated with such a tool.
- **GPT-3** (Brown et al., 2020): GPT-3, with its 175 billion parameters and having been trained on 570 gigabytes of diverse data, demonstrated remarkable generalization and text generation capabilities. Two years later, GPT-3.5 brought this technology into the mainstream with the launch of ChatGPT. It is worth noting that from GPT-3 and onward, OpenAI has not open sourced their models, a decision that has drawn a lot of criticism.
- **GPT-4** (OpenAI, 2023): GPT-4, marked a significant step-up in language processing capabilities, introducing enhanced reasoning, contextual understanding, and the ability to interpret both text and images, making it the first in the GPT series to offer the capacity for vision. Soon thereafter, GPT-4o was released, providing incremental improvements in efficiency and performance. Reinforcement Learning with Human Feedback (RLFH) was incorporated in these models to train them to align better with human values. While the technical details remain undisclosed, these models are estimated to have more than a trillion parameters.

Since the first GPT was released, many researchers and companies have invested in LLMs, contributing to the rapid development in the sector. While a lot of the more advanced models remain proprietary, a growing number of them have been made available to the public. The tech giant Meta has become a major contender in the open source LLM effort, with their LLaMa (Touvron et al., 2023) models contesting OpenAI's GPTs in many benchmarks. Some other notable freely available options include Google's BERT (Koroteev, 2021), BLOOM (Le Scao et al., 2023) and Qwen (Bai et al., 2023).

The language generation and comprehension capabilities of these models is a difficult thing to adequately measure. Standardized benchmarks (Zellers et al., 2019) (Hendrycks et al., 2021) across various different tasks have been created to better facilitate this but it is still a heavily debated and controversial process in the LLM community. As a result, no single tool has emerged as a definitive winner, with the choice depending on many factors such as the nature of the task at hand, the required level of accuracy and computational resources.

Hugging Face (Jain, 2022) is currently one of the most popular platforms for researchers and enthusiasts interested in Natural Language Processing and Large Language Models. It provides advanced tools,

powerful libraries and APIs that can streamline the process of training, fine-tuning and deploying state-of-the-art models. It is an essential resource and will be a central part of the experiments conducted in this study.

Having established the key technical concepts that LLMs are built upon, ranging from the basics of ML to the more elaborate architectures and systems that power these models, we now focus on the specific domain where we aim to apply these technologies: academic research review. In the following chapter we examine the research process, exploring its key stages, challenges and intricacies, as well as the specifications for adapting LLMs to this highly specialized area, focusing on the analysis and classification of research papers.

Chapter 3

The research process and the role of LLMs

In this chapter, we examine the continuous evolution of science and underline how the development of new, increasingly powerful tools has altered the research process. Technological advancements have trivialized parts of the workflow that would have once required significant amounts of time and effort to complete, allowing the researchers to focus on more critical parts of their work. Artificial intelligence and particularly Large Language Models have taken an assistive role through many stages of research and are have already began to transform many aspects of the process. In this chapter, we will examine the evolution of science and research, the challenges currently facing the field, the integration of technology and the role that LLMs can play in peer review.

3.1 Science and research

Science represents the body of knowledge that humankind has attained over the years as well as the methods used to create and evaluate new knowledge. It is an ongoing, structured process founded on protracted observation, repeated experimentation and the subsequent formulation of theories which are continuously tested and refined. Science seeks to explain and predict aspects of the world through empirically and logically sound methods in order to achieve the expansion of human knowledge and the solution of real-world problems through practical applications.

The scientific process must strictly adhere to certain principles. Inquiry must be based on empirical data obtained from experimentation or observation. The methodologies and data used should be reported to the broader scientific community and findings must be consistently reproducible by other researchers under similar conditions, to ensure the reliability and accuracy of the proposed ideas.

Science is deeply embedded into society and inevitably influenced by economic, cultural and political factors. Ethical guardrails must exist firmly in place to ensure not only the accuracy of research but also the moral validity of the scientific process. Science must adhere to the same principles that apply to the rest of society: ”...honesty, fairness, objectivity, openness, trustworthiness, and respect for others” (National Academy of Sciences et al., 2009).

Ensuring the integrity of the scientific process is critical, however, it is equally important to recognize the impact that scientific discoveries and their applications may have on our society. Dual-use Research of Concern (WHO, 2020) refers to research which has the potential to be misused and cause harm, even

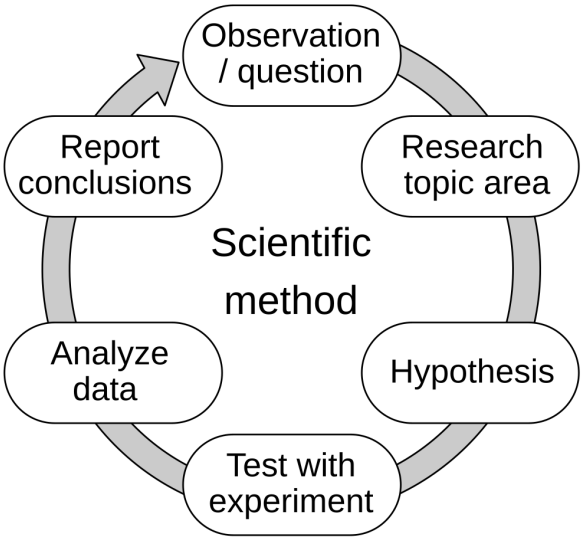


FIGURE 3.1: The scientific method (Efbrazil, 2021)

though its initial purpose is transparently beneficent. A topical example of this phenomenon is Artificial Intelligence. The potential of this technology to aid in the automation of mundane tasks, enhance existing systems in education, healthcare, agriculture etc. can be overshadowed by the very real threat of its misapplication through mass surveillance and disinformation campaigns, or even as means of war. A 2020 estimation put the number of bots on the social media platform X, formerly Twitter, at 48 million, or 15% of the entire user base (Rodríguez-Ruiz et al., 2020). The efficiency with which large bot networks can influence the cultural and political zeitgeist is only to have grown following the large scale adoption of large language models.

With these considerations in mind, we now delve deeper into the research process and analyze it's evolution throughout the years, the challenges it faces, the manner in which it is evaluated and the role that technology can play in that process.

3.1.1 The research process

Research is a systematic approach to generating knowledge and answering questions. While the process is not rigidly defined and may vary from discipline to discipline, the core structure remains similar and follows several coordinated steps:

1. The research begins by identifying a question or a problem in need of a solution. This includes a literature review to locate gaps in the field that can be explored, thus deciding on a topic and formulating the central subject matter.

2. A blueprint is designed for how the research will be conducted. This may involve selecting methodologies, defining the requirements and the desired outcomes as well as going over possible ethical concerns.

3. Data is collected through means of experimentation and rigorous testing. Information may also be sourced from previous studies and investigations. It is paramount, in this step, to ensure the validity and reliability of all data extracted.
4. Once the data is collected, it can be analyzed to interpret patterns and test the underlying hypothesis of the research effort.
5. Following thorough analysis, the researchers may reach conclusions either prove or refute their initial theory. Importantly, they must acknowledge the constraints in their study and also consider the broader implications of their findings.
6. When the study is finished and results are verified, the findings are shared with the broader scientific community, by presenting them at conferences and publishing them in journals, where they will be assessed by independent experts in the field of and judged based on their validity and novelty.
7. Research being an iterative, cyclical process, the final part entails reflection over the findings and their implications, assessing how they address the original research hypothesis. From this, new questions may arise, laying the groundwork for future research and beginning the cycle anew.

3.1.2 Challenges in modern research

Research methodologies have evolved immensely throughout the course of humankind's pursuit of scientific knowledge. From the logically driven, deductive reasoning traced back to ancient Greek philosophers to the scientific revolution during the Renaissance which emphasized empirical evidence and systematic experimentation as pillars of the scientific process, paving the road for modern research. In current times, computational tools have radically changed the way research is conducted once again, with artificial intelligence representing another significant advancement.

Despite all the impressive improvements in methodologies and technologies, modern research still faces many challenges that can dampen the efficiency with which it is produced and the impact it has on society and academia. In order to be able to address and mitigate these problems, they must first be clearly outlined and comprehended.

The replication or reproducibility crisis plagues a large part of modern scientific discoveries. Findings show that a significant proportion of researchers - more than 70% - have attempted to reproduce another scientist's experiments and failed, with many also having failed at replicating even their own work (Baker, 2016). Rates on a field by field basis can be seen in Figure 3.2. That is not to say that more than two thirds of all published research is false. It does, however, bring into question the reliability of many findings and whether the research was conducted in good faith, without bias or alteration of results, and following all due processes. The most common factors that lead to irreproducible research cited where selective reporting, a pressure to publish and, simply, inadequate statistical analysis.

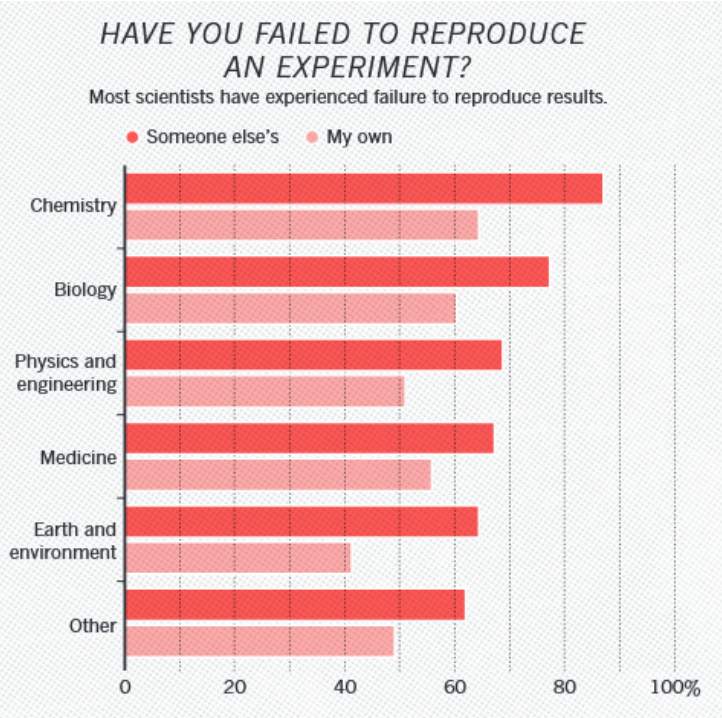


FIGURE 3.2: Reproducibility failure rates (Baker, 2016)

Bias, which occurs when design choices, analytical methods and selection of data result in findings not supported by the evidence, is a key component of inaccuracy in research (Ioannidis, 2005). Under a biased framework of thought, a researcher may draw conclusion errantly, subconsciously creating relationships in data that do not exist. Due to the nature of science, this is not an issue that exists in isolation. A new study that is based on older research will inadvertently be affected by this bias, even if it was itself conducted perfectly. This can result in subtle fallacies being compounded into academic literature, affecting the field for years to come.

The struggle to constantly output new research in order to retain relevancy and, more importantly, funding, exacerbates a lot of these issues. The culture of ”publish or perish” can lead researchers to cut corners and not be as thorough in verifying their results, or, in the worst of cases, purposefully embellish their findings and misrepresent the truth. This creates a highly competitive and destructive ecosystem that dissuades cooperation and encourages unethical behaviors.

Another limitation of modern research is the quality and accessibility of data. A prime example is OpenAI's refusal to share the corpus on which their newer GPT models were trained. While this represents an edge case where even with access to the data it would be impossible for individual researchers or groups to reproduce such large scale experiments, a concerning precedent is clearly set. Without transparency, both for the methods used to collect the data and for the data itself, verification and replication become nearly impossible.

3.1.3 Evaluating the quality and influence of research

To evaluate research, various criteria can be used to gauge its credibility, impact and contribution to the field. There are quantitative metrics, such as traditional bibliometrics which measure citations or references between works, or altmetrics, which track online engagement with the work (University of Sheffield Library, 2024). They represent the attention that a paper, author or journal has received has received; not necessarily its quality. Author-level metrics track the overall influence of an researcher's complete body of work while journal-level metrics measure the prestige of an entire journal. Both focus on collections of work rather than the merit of individual papers.

Author-level metrics

- **h-index**: The h-index (Bornmann and Daniel, 2007) is calculated as the number of papers where the author has been cited by others at least as many times. An h-index of 5 means that the author has published at least 5 papers that have been each cited at least 5 times.
- **g-index**: The g-index (Egghe, 2006), proposed by Leo Egghe, states that if a set of articles "...is ranked in decreasing order of the number of citations that they received, the g-index is the (unique) largest number such that the top g articles received (together) at least g^2 citations". For instance, a g-inex of 10 means that the top 10 publications of the author have been cited collectively at least 100 times.

Journal-level metrics

- **Impact Factor**: The impact factor of a journal (Clarivate Analytics, 2024) is defined as the mean number of citations of the articles published in a given journal in the previous two years. It is calculated and published by Clarivate's Web of Science.
- **Eigenfactor**: The Eigenfactor (Bergstrom et al., 2008) is another rating of the importance of a scientific journal, measured by the number of incoming citations, with the added step of factoring in the importance of the journal of the incoming reference, measuring not only the quantity but also the quality of the citations.

Altmetrics offer a more modern alternative or complementary overview to classic bibliometrics. The examine less scholarly interactions with the research, tracking diverse forms of engagement in non academic settings. Common altmetrics (Bornmann, 2014) include mentions and likes in social media, blog posts and discussions around the paper, views, downloads, coverage by legacy media and more.

Altmetrics and citation-focused metrics serve distinct purposes. Citations capture the interest and impact in the academic community while altmetrics measure the real-time engagement of the public with the research. They can and should be used in tandem. They both provide a useful, albeit limited numerical indication of the significance of a paper.

Over-reliance on quantitative metrics, however, can be problematic as they tend to reflect popularity and connections rather than the pure quality of the research. They incentivize the publication of

many smaller articles to boost numbers rather than fewer more meaningful ones and encourage unethical practices to artificially inflate citation counts. They also potentially can transform the evaluation of research into a popularity race, where individuals with greater connections can come ahead at the expense of lesser known researchers who may even produce higher quality publications.

In an attempt to encourage academics to focus on a qualitative assessment of the literature rather than relying solely on quantifiable metrics, a set of principles for evaluating research have been proposed (Hicks et al., 2015):

- Quantifiable metrics should support qualitative evaluation, not the other way around. They are an extra tool that may help reduce personal biases but they should not be the main focus.
- Performance needs to be measured against the specific research goals of the institution, group or researcher. Not all research has the same purpose and the way it is judged should reflect that.
- Excellent research in niche fields will unavoidably garner less interest and citations; that does not mean it is not worthwhile and should be ignored.
- The way data is collected and analyzed for the purposes of evaluation should be kept transparent and readily available, enabling scrutiny and deeper understanding of the process.
- The researchers evaluated should have the chance to verify the accuracy of the presented data in any given bibliometric study that includes them.
- It is crucial to consider the different publication and citation practices on a field by field basis, as the standard can vary a lot and inadvertently penalize certain disciplines.
- Individual researchers should be judged based on a qualitative evaluation of their output, as metrics tend to uplift older researchers and trending fields.
- High precision and confidence in uncertain metrics is unwarranted. They are highly debated concepts constructed to help numerically gauge research and should not be treated as unquestionable fact.
- It is critical that the systemic effect of metrics on the state of research are recognized, as they can incentivize anti-scientific publication habits in order to maximize appeal and funding.
- Metrics and indicators must be frequently assessed and updated to reflect the changing nature of research, as well as the culture and goals surrounding it.

These principles are not a rejection of common statistical metrics, as they are an important and objective assessment of the outreach of a publication. Higher citation counts and prolonged engagement with the public can certainly indicate a higher quality paper with broad implications and impact. They simply represent a warning for scientists and people in academia to not depend entirely on numbers and seek a deeper understanding of the literature they read.

A more thorough, qualitative assessment, requires a reviewer with prolonged experience in the field and up do date knowledge of the latest research developments. Qualitative metrics focus on the depth,

originality and impact of the research rather than quantifiable indicators. However, there is no single definitive method for evaluating research papers in this manner. The process is often subjective, tailored to the specific disciplines, contexts, and preferences of the reviewer.

The following set of criteria, while not exhaustive or definitive, outline some of the essential aspects of research (Tracy, 2010):

- **Worthy topic:** The research must address an issue that is relevant, significant, or intriguing, engaging with problems that concern both the academic community and society in general.
- **Rich rigor:** The study uses high quality, appropriate and complex theoretical constructs, data, samples, analysis processes etc.
- **Sincerity:** The values, biases and inclinations of the researches must be recognized to contextualize the study. The methodologies used and challenges faced must be readily apparent.
- **Credibility:** The research must be reliable and believable, featuring in-depth descriptions to help convey deeper meanings, sufficient immersion in the research topic and inclusion of many perspectives.
- **Resonance:** The study's topic and findings must be presented in a way that adequately impact the reader.
- **Significant contribution:** The study should result in insights that advance theoretical understanding, deal practical challenges, or create new avenues for research in the field.
- **Ethics:** Ethical considerations should guide every stage, from collecting the data to conducting experiments and drawing conclusions.
- **Meaningful coherence:** The study must be logical and consistent, connecting its research aims, methods, findings, and conclusions coherently.

3.2 Digital tools in research

Technology has radically changed the way research is conducted and evaluated, facilitating easier access to tools and platforms that can streamline the process and increase productivity. This evolution has accelerated the pace with which scientific discoveries can be achieved and shared as well as lowered the barriers to entry, enabling individuals and smaller teams to compete with larger established organizations without an unreasonably high cost.

The need to manually parse large datasets has mostly disappeared, as programming languages such as R, MatLab or certain Python libraries have emerged as an effective solution for analyzing data and simulating research conditions. They can instantaneously scan large quantities of information, find patterns or anomalies. This can be used to identify gaps in understanding and formulate new research propositions.

Collaboration in research has become easier and more accessible than ever before. From something as simple as Zoom, which has eliminated the need for travel for global cooperation, to specialized online collaboration tools like Overleaf, which enables the seamless cooperation of many writers and the capability for real time correction, improvement and critique.

The internet has also given rise to numerous pre-print repositories and open access journals. In 2024, an average of 20,000 papers were uploaded to arXiv each month (arXiv, 2025), as seen in Figure 3.3. These developments help researches share their findings before they undergo a formal peer review, allowing others to gleam insights and build upon their findings in a rapid manner. Open access publishing guarantees that papers and articles are freely available and easy to reach for all, not hidden behind paywalls. This particularly benefits researchers with limited means and serves to even the playing field.

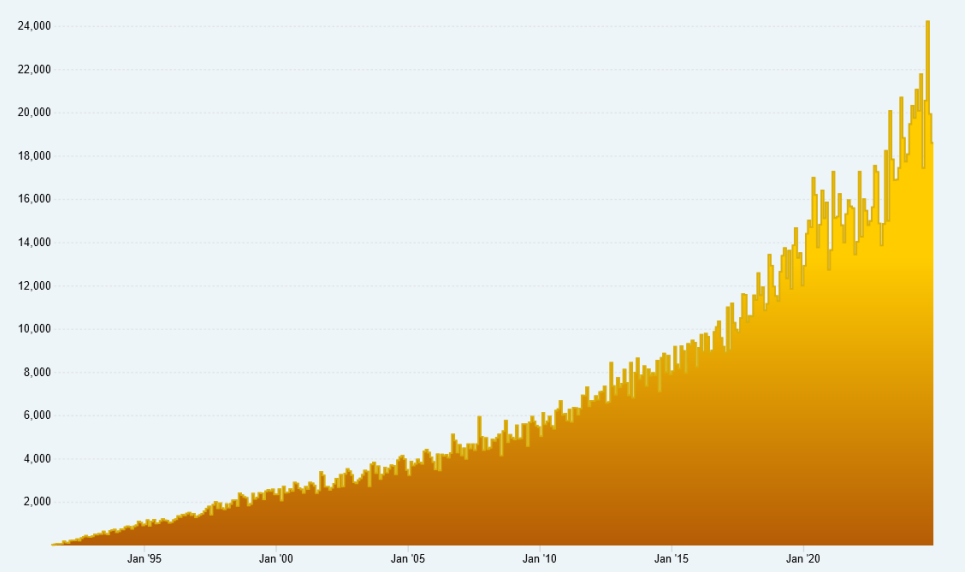


FIGURE 3.3: arXiv submission statistics (arXiv, 2025)

Platforms like OpenReview take this a step further by encouraging openness and transparency in the peer review process. Reviews on the platform are open and visible to all, fostering an environment that promotes constructive feedback and reduces the possible biases of any individual reviewer. The exact structure of the reviews differs from venue to venue but they generally include the perceived strengths and weaknesses of the paper, specific points of contention, requests for changes as well as questions posed towards the authors, who then have the ability to elaborate on their work and provide rebuttals. These steps may be repeated until the reviewers are satisfied and their concerns have been addressed, at which point the paper is accepted for the venue. Otherwise, the paper is rejected.

The combination of pre-prints, open access models and transparent reviews have fostered sizable online communities focused on sharing, discussing and expanding on findings, by open-sourcing the process. They have reshaped academic writing and publishing, improving inclusiveness and collaboration, and ultimately raising the quality and impact of scientific research.

Automating the writing, analysis and critique of research papers, as well as other academic forms of writing, has gained significant traction with the creation of new and sophisticated software. These tools aid researchers, reviewers and educators by offering objective assessments and meticulous examination.

Writing assistants

Grammarly (Fitria, 2021) and Writefull (Tech to Words, 2023) are a set of AI enhanced writing assistants that utilize deep learning and natural nanguage processing techniques. Through their use of language models models they can achieve much more than simply detecting spelling mistakes; they cover a broad array of syntactic and logical errors. They have the ability to recognize the style and tone of the writing and suggest changes that will produce the desired effect. Writefull is specialized specifically for scientific literature and has been trained on a large corpus of academic work. The exact processes and algorithms used by both of these applications are proprietary and not readily available to the public.

Plagiarism/AI detectors

Turnitin (Batane, 2010) and iThenticate (McCulloch et al., 2022) are plagiarism detection systems used widely by academic institutions to identify content similarity between assignments and papers and the existing literature. Text is examined against a massive dataset and ascribed a similarity rating. This is achieved through the use of fingerprinting techniques (Turnitin, 2016). Rather than trying to match keywords or strings of words exactly, it ascribes a unique fingerprint to the works based on their style, tone and phrasing. This provides the ability to detect not only obvious cases of plagiarism but also more subtle, poorly paraphrased instances. With the rapid developments in the LLM sphere, these systems have also had to introduce AI detection capabilities in order to track writing generated by GPT models.

Research assistants

Elicit (Kung, 2023), Semantic Scholar (Fricke, 2018), Scholarcy (Bui and Bui, 2024) and Zotero (Mueen Ahmed and Dhubaib, 2011) are AI-driven tools designed to assist researchers at different stages their academic work, each serving unique purposes. Scholarcy focuses on summarizing papers and extracting key points and highlights to aid in understanding complex content faster. Elicit facilitates systematic literature reviews and can answer research questions by going through relevant papers and extracting information, as seen in Figure 3.4. Semantic Scholar is a search engine for academic writing that can analyze citation contexts and show the influence of papers. Zotero is a reference management tool that can complement the above systems and help researchers collect and organize their citations through their extensive database. Together these tools support the author through the entire writing process and can help produce higher quality research.

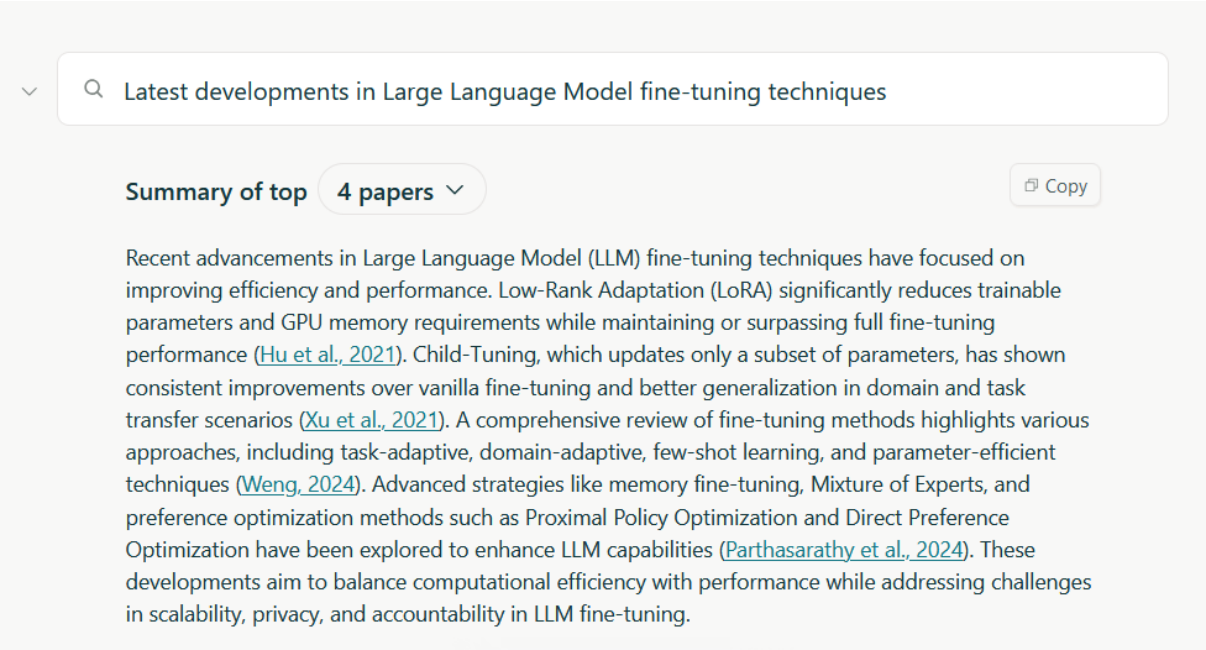


FIGURE 3.4: Elicit AI search and summarization (Elicit, 2023)

These tools generally focus on singular, pre-defined tasks and their purpose is to assist the researcher with the more repetitive and laborious aspects of their work. They are unable to critique a paper holistically, evaluating it's significance and novelty, especially when it comes to areas that require thorough understanding of complicated subjects.

3.3 AI models in academic reviews

Automating the critique of research papers faces several important challenges. The language used in them is generally more complex and full of nuanced arguments and complicated theoretical concepts. Additionally, the subject and field of the paper would alter the methodology used to examine it. For instance, humanitarian studies would be held against different evaluation criteria compared to Science, Technology, Engineering, and Mathematics (STEM) fields. And, crucially, assessing whether a contribution is original and topical can be extremely difficult. Even if a model where to maintain up to date knowledge in the field, it would still require very good reasoning capabilities to be able to recognize novel and meaningful research.

Artificial intelligence, through machine learning and natural language processing, has the potential to transform the automated critique of research papers. By analyzing vast quantities of academic literature, AI can learn the key metrics that must be satisfied and how to identify them in a paper. A single human reviewer can keep up with a limited number of disciplines and the subfields within them. The rapid pace with which new research emerges demands a significant investment of time to not only comprehend the novel ideas presented but also to evaluate their overall significance.

Large language models, by training on large diverse datasets of academic writing, can learn to recognize the required structure of research papers, gauge the methodologies used and locate mistakes or inconsistencies. LLMs can provide an assessment of a study's originality, significance, and potential impact, offering insights to reviewers and editors. The combination of AI and human expertise can increase the speed of the process and reduce the effort that a thorough review requires, making the task of evaluation more efficient and scalable.

Training models that can adequately face these challenges requires a meticulous approach and careful planning. A diverse and balanced dataset that represents multiple writing styles and research methodologies across a variety of domains must be constructed. The data must be annotated with elaborate and extensive review criteria that indicate a paper's worth in a wide selection of subjects, such as the writing quality, the sufficiency of the experimentation or the novelty of the ideas. Additionally, ensuring that a model can understand very niche technical terms, specialized vocabulary and conventions necessitates extensive domain-specific fine-tuning.

Fine-tuning

The process of adapting a model to a specific domain by further training it on a carefully curated dataset is known as fine-tuning (Jeong, 2024). Fine-tuning is a crucial step in tailoring LLMs to better understand the requirements of a certain task and generate more accurate outputs that adhere to specific tonal and contextual guidelines. This process is exceedingly important when dealing with niche disciplines or cases where mistakes can be extremely costly, such as the medical or legal fields.

The traditional approach to fine-tuning involves updating all the weights of the original network. While effective, this process becomes increasingly resource intensive as the size of LLMs scales. Considering the sheer size of modern GPT architectures and the computational requirements of endeavoring to train them in their entirety, several Parameter-Efficient Fine-Tuning (PEFT) techniques have been proposed. Key examples include Adapters (Hu et al., 2023), Prefix-Tuning (Li and Liang, 2021) and Low-Rank Adaptation (LoRA) (Hu et al., 2021).

Adapters introduce small trainable layers to the model in-between existing layers. During the fine-tuning process, only the weights of the adapter layers are updated, while the rest of the network remains frozen. Pre-fix tuning works by adding a series of trainable prefixes to the input tokens at each layer, modifying the way in which the model interprets the data. Without having to alter any of the weights of the model, this method provides an easy, resource-cheap way to adapt models to different tasks.

Low-Rank Adaptation is a very popular option in the PEFT category. The full weights of the model are frozen and smaller trainable decomposition matrices are inserted into each layer of the Transformer. In essence, LoRA assumes that the update to the weights of the model can be approximated as the product of two matrices with significantly lower rank than the original. Thus, we only need to train the parameters of those two matrices, drastically decreasing the amount of computation and memory required.

Quantization, pruning and mixed precision training

With the aim of further improving the performance of LLMs in terms of memory use, computational efficiency and inference speed without sacrificing the accuracy of the model significantly, numerous methods have been proposed. These developments have been a game changer for researchers and developers with limited resources, allowing them to locally train and test some of the more powerful modes.

Quantization (Jacob et al., 2017) is the process of reducing the precision of the model's parameters from their typical 32-bit floating points to lower precision formats such as 8-bit integers. This way the memory footprint of the model is markedly reduced leading to faster, more efficient inference. Quantization is most commonly done after training, where the trained weights are converted to the lower precision. Quantization-aware training can also be used, where the model is trained while accounting for the lower precision, which leads to a lower loss of accuracy when it is finally quantized and deployed.

Pruning (Ma et al., 2023) an LLM refers to removing parameters from the model which are deemed to not have a significant impact on the overall performance. By removing the unnecessary weights, it can notably decrease the memory required to store the model as well as the time required for inference. However, the process must be done carefully to ensure that essential features of the model aren't lost, which could lead to a big loss in accuracy.

Mixed-Precision Training (Micikevicius et al., 2018) is a technique which, similarly to quantization, uses lower precision for certain parameters and operations to lower the time required for training while not sacrificing performance. Operations where high precision is not as crucial are done with 16-bit floating points. However, for operations such as gradients or weight updates, where the lower precision could lead to numerical instability and cause catastrophic issues during the training, 32-bit floating points are typically used.

Retrieval-Augmented Generation

The constant outpour of new and highly relevant research would also necessitate the existence of an extensive database that is regularly updated with the latest developments and can serve as a baseline for the decisions of the model. The system must have the ability to examine the citations of the paper and determine whether the referenced works are relevant and also search for similar papers or publications and decide whether the work presented is a novel idea, an incremental contribution or something that has been covered before and is not of interest.

For this purpose, Retrieval-Augmented Generation (RAG) (Gao et al., 2024) can be applied to enhance the outputs of the LLM with access to dynamic knowledge. It combines retrieval based models with LLMs and is particularly effective for tasks where factual accuracy and specific knowledge is heavily required. RAG models consist of two parts, a retriever and a generator. The retrieval component scans through a large corpus to find relevant information based on the query. Once the relevant documents

have been found, the generative model uses the information in them to generate a more accurate response.

RAG is a way to incorporate real-world data in real-time, making it more dynamic and adaptable while remaining grounded in fact. Unlike traditional models trained on static dataseds, RAG models that are constantly provided with new information can evolve their knowledge and remain up-to-date. Rather than needing to have all knowledge embedded and memorized they can only retrieve what is deemed necessary, making them a more scalable option.

Multimodal LLMs and context size

Multimodal LLMs (Yin et al., 2024) are a class of models that are able to process data in multiple different forms, such as text, images, audio or video. They aim to create a better understanding by analyzing more diverse input data types. This is particularly important for the academic review process, as papers often feature many figures and diagrams. It is crucial to ensure that the quality of the images matches the rest of the paper and that the captions adequately describe the contents and provide necessary context.

Another important aspect to consider when selecting a model for any given NLP task is its context size. The context size of an model refers to the amount of tokens it can process at once. For a GPT model, it is the maximum amount of information that it can remember in one inference. This is an extremely important concern in regards to evaluating research papers, as they usually span thousands of words. Extremely powerful models would be necessary to process them all of the content at once. Various methods can be used to mitigate this issue such as parsing sections individually or summarizing previous sections and working with the summaries.

Reinforcement Learning with human feedback

Leveraging the knowledge of experts through Reinforcement Learning with Human Feedback (Lambert et al., 2022) can also play an important role in ensuring the quality of such systems. The decisions and thought processes of reviewers can help shape the tone and improve the organization and flow of the reviews. Moreover, it adds an extra layer of human evaluation for the model that can serve to alleviate the concerns of users by having people with extensive knowledge of a field verify and approve the performance.

Explainability

A general concern with deep learning models is always the explainability and transparency behind their decisions (Zhao et al., 2023). They are often regarded as a ”black box” whose process for reaching a certain output cannot be fully understood. This can create doubts as to the quality or accuracy of such systems. That is why an important part of such a model must be the ability to adequately rationalize its decisions and substantiate its claims. This is paramount to ensure that it acts without inherent biases and provides a fair assessment across many disciplines and subject matters.

As with any new and exciting prospects in the field of artificial intelligence, the ethics of such systems come into question. Ensuring that the models remain unbiased and accurate requires careful and deliberate design choices and thorough pruning and management of the data they are trained on. It is equally important to recognizing the limitations and the potential risks of relying exclusively on automated tools. A balanced approach is needed to make the most of these new technologies while still protecting ourselves from unintended consequences.

3.4 Insights from code review automation

Automating the majority of the software engineering process has become one of the prime areas of interest for large language models. Their ability to generate code from scratch or improve and build upon provided code has rapidly grown, transforming them from an unreliable source to a trustworthy tool. The capacity to understand and analyze code suggests that, in theory, they could also function as an automated code reviewer.

Code review and academic review largely share similar goals. To detect errors, improve the overall quality and enforce certain standards. They both require strong critical thinking and a solid understanding of the entire underlying structure and purpose of the work. Through this illustrative example we can gather insights and techniques that transfer over from one endeavor to the other.

In code review, LLM's can identify structural issues in parts where the flow of the code is unclear or poorly sectioned, such as deeply nested loops or massive chunks of repeating code not turned into functions. In a paper, this is could be akin to an imprecise laying out of the methodologies used or a lack of logical flow from the findings to the conclusions of the paper. The LLM assistant in either case would recommend reorganizing entire sections to lead to a cleaner final product.

Furthermore, maintaining consistency is paramount in both academic writing and software engineering. Code has to adhere to the specific style guide adopted by the entire codebase. Violations such as different indentation or improper variable names can cause issues many issues in the future. Similarly, references in a paper must follow an exact citation style and be formatted in a specific layout.

Finally, detecting clear-cut errors is possibly the most important and difficult responsibility for such systems to perform. For code review, this could refer to errors that would lead to an overflow, infinite loops or other issues in the logic of the code that would not be revealed by a compiler but would nonetheless lead to incorrect results. In academic reviews, they system would flag obvious logical fallacies or errors in the statistical appraisal of the findings. This step requires the strongest skills in critical evaluation and prolonged expertise in either field.

These parallels serve to showcase the potential carryover from one effort to the other. They both concern relatively structured forms of writing without excessive room for deviation, where subtle discrepancies can have a critical effect on the overall quality. Their review process are both iterative and focused on repeatedly finding and remedying mistakes until the final product adheres to a high enough standard.

Large language models have shown promise in automating the code review process (Rasheed et al., 2024). By analyzing entire repositories of code they are able to identify a variety of issues, from minor bugs to significant overarching inefficacy and architectural flaws. A multi-agent approach has been explored, with specialized components for code review, bug detection, code smells, and optimization. The training dataset consists of code repositories from Github, including code reviews, bug reports and best practices from documentation. The agents were tested in a range of AI applications that cover different domains and technologies. The model's potential as an educative tool that provides developers with a better understanding of the code and its issues is also examined. While the current findings are only an early evaluation, they are an informative view into the potential of LLM-based systems for the automation of the code review process.

Comparably, empirical research has investigated the prospect of using LLMs to detect defects in security during the code review process (Yu et al., 2024). Numerous different LLMs have been tested and had their performance measured against static analysis tools on datasets consisting of review comments that identify security issues. Different prompts were also tested to to coax specific information out of the models. The evaluation was conducted across four main axes: correctness, understandability, conciseness and compliance, as seen in Figure 3.5. The final results show promise, indicating that the best performing LLMs outmatched the static tools. However, their use remains limited as the models are prone to unnecessary verbosity and can often fail to comply with the exact requirements of the prompting.

These cases of LLMs being used for code review offer valuable lessons and ideas for automating paper reviews. An important insight is the use of fine-tuning techniques that provide the models with a better understanding of highly specialized terms that are specific to niche domains. They also serve to shape the tone and language used by the LLM to better fit the academic setting. For this, representative and balanced datasets must be carefully constructed and given into the model.

Another crucial lesson is understanding the need for human supervision. These technologies are still at a very early point and cannot be used reliably on their own. It is important to think of them as an extra tool in the arsenal of the reviewer rather than the solution to all of their work. This aligns with the need for these systems to be able to clearly articulate their points in a way that is understandable, concise and valuable to the user. Being able to explain why an aspect of the work they are critiquing is flawed is as important as being able to detect the flaws in the first place. The tools must, at least for now, function as assistive to the review process rather than attempt to eliminate the need for human experts.

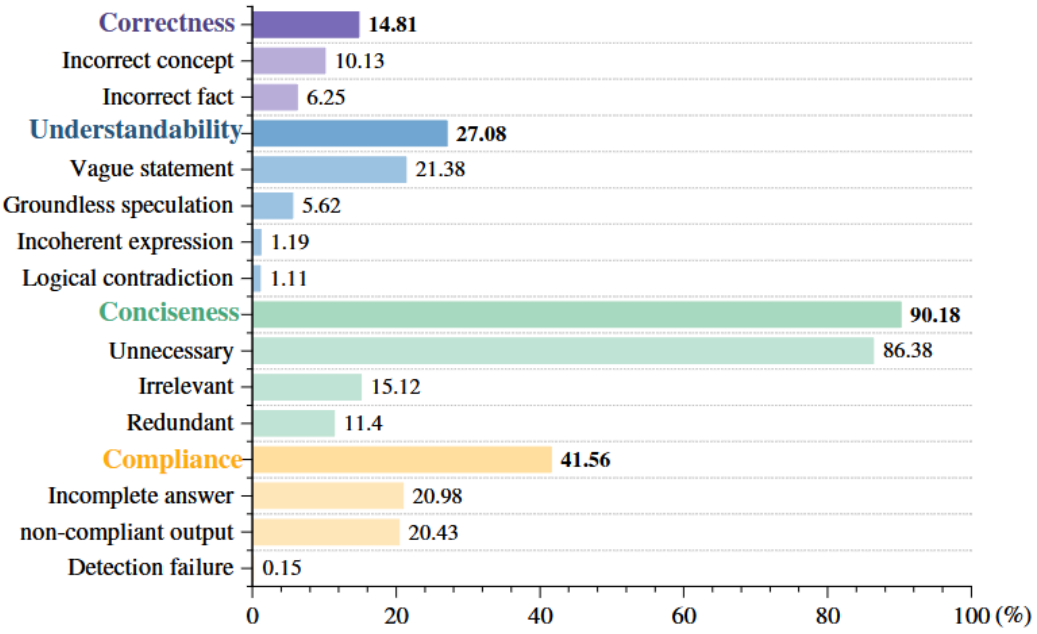


FIGURE 3.5: Common issues with LLM responses in security code review (Yu et al., 2024)

With the understanding of both the academic review process and the technical details of adapting LLMs to specialized domains, we begin the practical implementation of our system. The following chapter details the steps for building and deploying the system, from acquiring and preparing training data to selecting our models and fine-tuning them for maximum performance.

Chapter 4

Research paper review through Large Language Models

In this chapter the entire process of building the LLM-based paper review system is outlined, detailing all the stages from the collection and preparation of data, to the set up, training and evaluation of the various models that are used. The end goal of this system is to assist in academic peer review by analyzing papers and evaluating whether they are to be accepted based on their content.

4.1 Paper segmentation

The initial step of the automated reviewing pipeline entails the segmentation of the text into the many sections that comprise a research paper. The section categories were selected on the basis of being general enough to apply across domains and disciplines but also being able to provide a semantically meaningful split of a paper. These categories are: abstract, introduction, preliminaries and related work, implementation and methods, results and experiments, conclusion and discussion, references, other. The training of machine learning models for this type of task requires large quantities of high-quality data. For the construction of this corpus we utilize the OpenReview API to obtain the scientific documents and regular expressions to assist with labeling the dataset.

4.1.1 OpenReview data acquisition for segmentation

OpenReview's API serves as a valuable tool for accessing structured data from academic peer review instances, giving access to a range of metadata and content associated with many major and minor conferences. Each item in OpenReview is contained as a *Note*, a data structure that can represent various types of content such as paper submissions, comments, decisions and more. These notes all have unique identifiers and forum IDs, through which relevant information can be queried and accessed, such as all reviews under a paper.

The API has two major versions, API V1 and API V2, each with different structural conventions. Certain older conferences and submissions are accessible only through the old version of the API whereas newer ones are only available through V2. Comparability between the two versions is not maintained, making it necessary to detect the correct version dynamically based on the source of the data. Using a known note of a venue, the script tries both versions of the API and falls back to the available one.

```python
def get_convention_papers(convention_id, client, paper_limit=10, file_path="paper_pdfs", use_api2=False):

    papers = client.get_notes(invitation=convention_id, limit=paper_limit)

    papers_downloaded = 0

    for i, paper in enumerate(papers):
        if use_api2:
            title = paper.content.get('title', f'No title ({i})').get('value', None)
            pdf_id = paper.content.get('pdf', None).get('value', None)
        else:
            title = paper.content.get('title', f'No title ({i})')
            pdf_id = paper.content.get('pdf', None)

        if not pdf_id:
            print(f"Paper '{title}' has no PDF path.")
            continue

        pdf_url = f"https://openreview.net{pdf_id}"
        download_path = f"{file_path}{paper.id}.pdf"

        print(f"Processing: {title}")

        save_paper(pdf_url, download_path)
        papers_downloaded += 1
```

FIGURE 4.1: Paper acquisition through the OpenReviewAPI

Once connected, the script acquires the papers of the given venue through the function seen in Figure 4.1. For example, with a convention_id of ”NeurIPS.cc/2022/Conference/-/Blind_Submission”, the API would retrieve a list of notes from the NeurIPS 2022 conference. Each note contains data such as the title or abstract of the paper, but most importantly, a link to the PDF that contains the full document. For the segmentation training pipeline, only the PDF files are downloaded and saved.

4.1.2 PDF parsing and paragraph-level labeling for segmentation

With the PDFs downloaded, the next step entails structuring their contents in a form that can be used for ML model training. The processing of the scientific documents was automated using the PyMuPDF library (Raute and Contributors, 2024), which provides tools for accessing and extracting text from PDFs. The script iterates over a specified directory, accessing each document individually and commits the extracted content into corresponding CSV files. The paragraph division is handled by PyMuPDF, which identifies and returns the blocks of text found on every page.

For every document, the script records three key attributes per block: the raw paragraph text, a numerical value that represents the normalized order of the paragraph’s appearance in the paper and a section label, which is initially generated through regular expressions. These entries are committed to the CSVs and are subsequently manually relabeled if it is required.

The regular expression labeling process serves as a lightweight way to perform a basic segmentation based on common patterns in the text. Specifically, two categories of patterns were used. The first pattern looks for sections commonly found in scientific publications that typically are not preceded by a numerical prefix, such as the ”Abstract”, ”References”, ”Ethics Statement” etc. The second pattern is designed to identify numbered section titles, such as ”1. Introduction”, ”2 Methods”, ”1.1 Related Work”,

by matching prefixes followed by section names. The expressions, which help generate a preliminary segmentation for the document, can be seen in Figure 4.2

```
patterns = [
    # Looks for the sections that are are not usually accompanied by a number
    r'(?i)(ABSTRACT|REFERENCES|BIBLIOGRAPHY|ETHICS STATEMENT|REPRODUCIBILITY STATEMENT|AUTHOR CONTRIBUTIONS)',

    # Looks for number-word pairs: 1 Introduction, 2. Methods etc.
    r'(?:^|\n)(\d+(?:\.\d+)*\.?)\s*\n?([A-Za-z&\-][A-Za-z&\s\-]*)(?=\n|$)',
]
```

FIGURE 4.2: Section header detection with Regular Expressions

This approach faces certain limitations, which is why it is only used to assist with labeling rather than being part of the completed segmentation pipeline. Regular expressions are sensitive to format variation and inconsistency, both of which are common issues that occur when parsing PDFs for text. Additionally, equations and numerical sections tend to generate a lot of false-positives and create highly incorrect splits in the paper, which would be catastrophic for the final model. To create an adequately robust papers segmentation process, machine learning for natural language processing is utilized.

4.1.3 BERT and LSTM hybrid model for segmentation

To address the limitation of regular expressions, the segmentation pipeline utilizes a hybrid model architecture built on top of the Hugging Face Transformers library and Pytorch. The architecture integrates a pretrained BERT model with a sequential LSTM layer. Specifically, SciBERT (Beltagy et al., 2019), a transformer model pretrained on multi-domain scientific publications is used to extract contextual embeddings from each paragraph that can accurately represent the core meaning of the text.

The flow of the ParagraphClassifier combined model, whose architecture can be examined in Figure 4.3, is the following: Each paragraph that is extracted from the paper is tokenized using Hugging Face's SciBERT tokenizer and passes through SciBERT. The outputs is a representation of the text that serves as a summary of the entire paragraph. Following the BERT model, the contextual embedding, along with the positional value of the original paragraph is inserted into a linear layer which outputs the logits corresponding to the predefined section classes.

In order to not treat paragraphs as individual units but as parts of a whole, the model continues by passing the logits, the embedding and the positional value through a bidirectional LSTM, whose outputs go through another linear layer and give the final predictions of the model. By providing the LSTM with the predictions of SciBERT and features of the raw paragraphs the intention is for the model to be able to capture paragraph to paragraph dependencies.

During training, the optimization of the model is done through a combined cross-entropy loss function, which takes into account both the predictions made by both the preliminary BERT model and the final LSTM classifier. These two values are then aggregated with a greater weight on the LSTM output - specifically a 70% to 30% split-, forcing the model to prioritize the conclusive predictions.

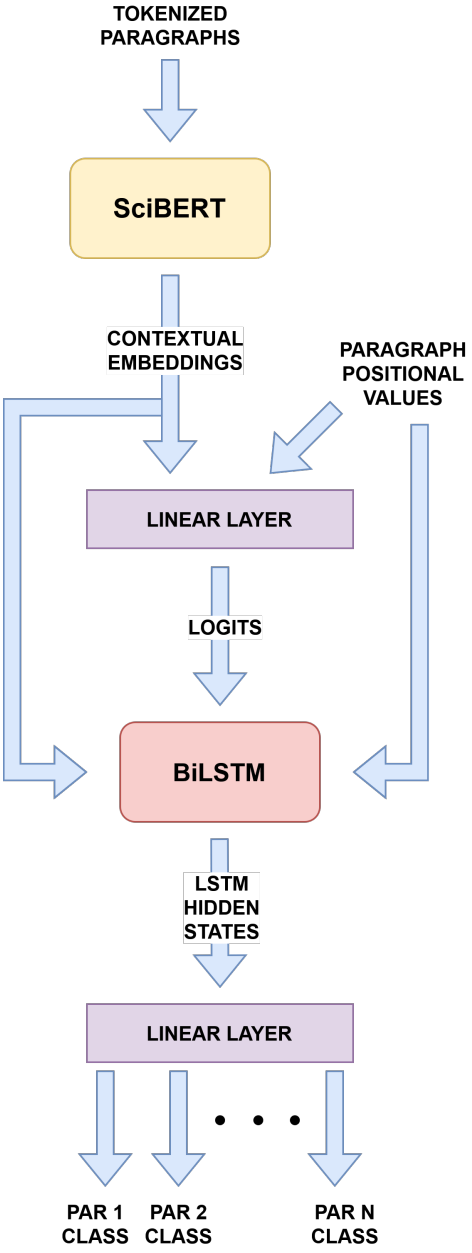


FIGURE 4.3: BERT and LSTM architecture

To prepare the data, a class inheriting Pytorch's Dataset is designed to process a collection of CSV files, each representing one paper. When an instance of the class is created it receives the folder where the files are stored, the name of SciBERT's tokenizer, the maximum number of tokens per paragraph and a flag for whether it will be used for training, in which case labels are also processed and a LabelEncoder from the scikit-learn library is used to ascribe integer numerical values to the different labels consistently across all documents and paragraphs. Additionally, a custom collate function is used to pad all documents of a batch to the same length to allow for efficient GPU-based parallelization.

4.1.4 Segmentation model performance

To evaluate the performance of the proposed model, experiments were run on a dataset of scientific papers segmented by section labels. Standard metrics, such as accuracy, precision, recall and more were used to were used to obtain a detailed breakdown of the all-around performance of the model across all classes. These were calculated with the scikit-learn library's classification report method.

Section Label	Precision	Recall	F1-score	Support
ABSTRACT	0.98	0.95	0.96	41
INTRODUCTION	0.85	0.93	0.89	246
PRELIM/RELATED	0.53	0.37	0.44	198
IMPLEMENTATION/METHODS	0.85	0.79	0.82	1386
RESULTS/EXPERIMENTS	0.84	0.68	0.75	1062
CONCLUSION/DISCUSSION	0.97	0.76	0.85	157
REFERENCES	0.99	1.00	0.99	846
OTHER	0.83	0.97	0.89	2154
Macro average	0.85	0.81	0.83	6090
Micro average	0.86	0.86	0.86	6090

TABLE 4.1: Segmentation performance metrics.

The segmentation model achieves an overall accuracy of 86%, indicating a strong capability for classifying the components of the academic papers. A more detailed overview can be examined in Table 4.1, which provides detailed information into the performance of the model across different sections. The micro and macro average F1-scores of 0.86 and 0.83 respectively further support the model's predictive capacity, suggesting robust performance not only across the more frequently occurring section labels but also across the less frequent ones.

In regards to the per-class performance, the model reliably identifies several key sections such as the REFERENCES, ABSTRACT, INTRODUCTION and OTHER. Each of these classes achieves an F1-score of 0.89 or higher, reflecting both high precision and recall. This is expected, as these sections tend to conform to well established formatting and linguistic patters have less variation from paper to paper. The structural and semantic similarity allow the model to better identify these classes.

Performance is notably weaker for the PRELIM/RELATED class, with an F1-score of only 0.44, primarily due to the weak recall of 0.37, suggesting that a large portion of these sections are being overlooked. The challenges with this class very likely occur due to overlap in the features and characteristics with other sections. Additionally, The low appearance rate of this label within the dataset in comparison to the the rest also likely exacerbates this issue.

For the IMPLEMENTATION/METHODS, RESULTS/EXPERIMENTS AND CONCLUSION/DISCUSSION sections, the model performs moderately well, with F1-scores of 0.82, 0.75 and 0.85 respectively. The numbers

suggest that while the model has a generally reliable performance, there remains certain room for improvement, especially in cases where content may deviate from the typical conventions.

To identify patterns in the model's incorrect predictions and get a better understanding of the underlying causes of it's misclassifications, we refer to the confusion matrix in Figure 4.4. This visualization provides valuable insight into the core issues of the model. Notably, the matrix indicates that a large part of PRELIM/RELATED sections tend to be misclassified as IMPLEMENTATION/METHODS. This makes sense, as a lot of information that is typically stated as preliminary in certain papers, such as descriptions the datasets, tools or models used, is sometimes included in the methodology in other papers. Due to this, paragraphs that are semantically and positionally indistinguishable can belong in different classes, posing a significant challenge for the model.

In summary, the model shows promising overall performance, particularly in the more clearly defined sections with less varied structure, such as the ABSTRACT or the REFERENCES. However, further improvement could be achieved for the more complex and variable sections, such as PRELIM/RELATED, where semantic overlap and structural inconsistencies present a greater challenge.

FIGURE 4.4: Segmentation model confusion matrix

4.2 Paper classification

In the second stage of the architecture, each document section is aggregated into a single, continuous text block. The aggregated paragraphs are then passed through a classification component, whose predictions are then aggregated to determine whether the section, and thus paper, is to be accepted or rejected.

4.2.1 OpenReview data collection for classification

Similarly to the data collection process for segmentation, the script gathers all papers from a given convention in the form of notes. Each of these notes can provide access to various associated sub-notes, which include the various steps of the peer review process, such as official reviews, author rebuttals, and, most importantly, the final decision note. The sub-notes are contained within the forum thread tied to the original submission.

The critical part of the process is the retrieval of the decision note associated with each paper. This is done by querying all the notes under the submission with an invitation string which, depending on the API version, follows one of the two following structures:

1. <Venue>/Paper<submission_number>/-/Decision
2. <Venue>/Submission<submission_number>/-/Decision

If a decision note is found, the verdict is extracted from the decision field as text. Then, any string containing the substring ”accept” is sorted to the positive acceptance label and similarly, the ones containing ”reject” are mapped to the rejection label. If the decision is unavailable or does not conform to either of these patterns the paper is omitted from the dataset. Additionally, the script enforces a balancing constraint to ensure an equal distribution of accepted and rejected papers. If the number of either type of papers falls behind the other significantly, papers of that type are skipped until the dataset is balanced again.

Once the decision note has been classified and validated, the paper’s PDF is downloaded and saved into a directory structure that is organized by class. Accepted papers are saved into the accepted directory and rejected ones into the rejected directory. This is done to make the dataset labeling easier and to keep track of how many papers of each category we have.

4.2.2 PDF processing and paragraph aggregation for classification

The PDF documents are processed analogously to the earlier step for segmentation and are split up into blocks, as defined by the PyMuPDF library. Once separated, these blocks are used as input to the pretrained segmentation model. The model evaluates each block separately and assigns to them their predicted class.

While the model is fairly accurate, inconsistency or noise can still affect the output, particularly when dealing with non-standard formatting or ambiguous transitions. To mitigate this issue, two different smoothing techniques are applied to the predictions:

1. **Short run filter:** This approach identifies blocks labeled with a section type that only appears briefly, i.e. for fewer than a specified number of consecutive blocks, and has on either side of it a different, consistent type. In such cases, we assume the brief segment to be a misclassification and the outlier segment is replaced with the class of its neighbors.
2. **Sliding window with majority voting:** For further stability, a sliding window mechanism is also applied across all the blocks. Within each window, the most frequent section is selected and applied to the central block. This majority voting technique enforces smoother transitions and is very effective against isolated errors. As this method's smoothing can be quite aggressive with larger window sizes, its effect on the sections was monitored to ensure it did not negatively affect the overall segmentation performance.

```python
def filter_short_runs(labels, min_run_length=6):
    corrected = labels.copy()
    i = 0
    while i < len(labels):
        current = labels[i]
        run_start = i
        while i + 1 < len(labels) and labels[i + 1] == current:
            i += 1
        run_end = i
        run_length = run_end - run_start + 1

        if run_length < min_run_length:
            prev_label = labels[run_start - 1] if run_start > 0 else None
            next_label = labels[run_end + 1] if run_end + 1 < len(labels) else None

            if prev_label == next_label and prev_label is not None:
                for j in range(run_start, run_end + 1):
                    corrected[j] = prev_label

        i += 1
    return corrected
```

FIGURE 4.5: Short run filter implementation

After smoothing is applied, certain section types are filtered before the classification step. Specifically, sections from the references and onward are intentionally omitted. References and the supplementary material that often follows them usually consist of bibliographical entries, appendices, disclaimers or checklists which offer little value for the purposes of content classification. By removing them we reduce noise for the model and ensure it focuses on the substance of the papers.

Once blocks are cleaned and smoothed out, all blocks that belong to the same predicted section are concatenated into one continuous text segment. This more coherent representation of the text is more suitable for the classification process. Additionally, each paper is associated with a label, either accepted or rejected, derived from it's OpenReview decision status, and this label is assigned uniformly to the section texts derived from it. The final dataset is comprised of structured, labeled samples, each one consisting with a large section of text and a binary decision label.

4.2.3 Classification model training

The model training process focused on fine-tuning the SciBERT model for the task of section-level classification. SciBERT's extensive pretraining on a large corpus of academic texts makes it well suited for this task. Each section of a paper is treated as an independent unit, with the goal of predicting whether it contributed to the final acceptance or rejection of the paper.

The dataset, as outlined earlier, consists of multiple CSV files, each containing the labeled sections of a single paper. Using Hugging Face's datasets, they are combined into a single dataset. Each section is comprised of its raw text and the corresponding class, either accepted or rejected. These text labels were mapped to binary values, 0 for rejected and 1 for accepted. The tokenization of the text is done with SciBERT's tokenizer, with the necessary steps taken for truncation or padding.

Once the dataset is properly tokenized and labeled, it is randomly split into train and test sets with an 80/20 ratio. The model is then fine-tuned with the Hugging Face Trainer API, which manages the training loop and evaluation strategy. Notably, fp16 mixed precision training was used, which accelerates training and conserves memory. The different training parameters and evaluation approaches used can be examined in Figure 4.6, which shows the training process through Hugging Face's Transformers and Trainer.

```
training_args = TrainingArguments(
    output_dir=output_dir,
    eval_strategy="epoch",
    save_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=64,
    per_device_eval_batch_size=64,
    num_train_epochs=3,
    weight_decay=0.01,
    load_best_model_at_end=True,
    logging_dir=os.path.join(output_dir, "logs"),
    fp16=True,
)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_train,
    eval_dataset=tokenized_test,
    tokenizer=tokenizer,
    compute_metrics=compute_metrics
)

trainer.train()
```

FIGURE 4.6: Transformer training through Hugging Face

4.2.4 Classification model evaluation

Once the training is completed, the first stage of the evaluation focuses on the model's section-level performance. This evaluation measures the model's ability to correctly classify the individual components of a research paper, as either accepted or rejected. This is the most immediate analysis of the model's predictive capacity, as it represents the task it was directly trained for.

The classifier achieved an accuracy of 77% on the test set, demonstrating a good ability for distinguishing between accepted and rejected sections of academic papers. Its precision of 72% and recall of 87% indicate that the model is more conservative with its ”rejected” predictions, opting more often than not to accept the section. The resulting F1-score of 79% reflects a well balanced trade-off of the two metrics, suggesting that the model can be expected to reliably identify the reception for a given section with a relatively high degree of confidence.

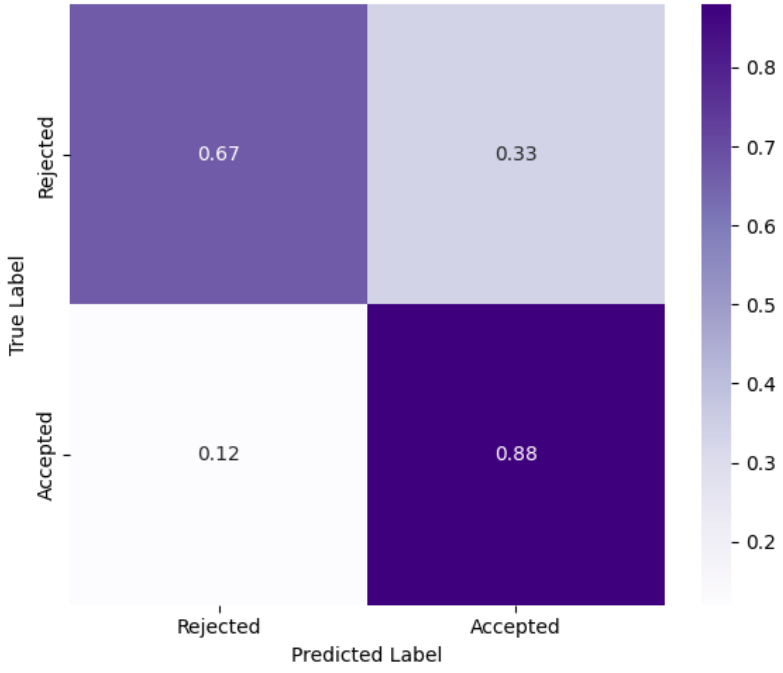


FIGURE 4.7: Section level classification confusion matrix

Further insight is provided by the confusion matrix in Figure 4.7, which reveals patterns in the model's classification behavior. For the sections that were truly accepted, the model correctly classified 88%, with only 12% misclassified as rejected. However, for sections that were truly rejected, 67% were correctly identified, while 33% were wrongly predicted as accepted. This further confirms that there is a certain degree of asymmetry to the model's predictions, with a certain bias towards the accepted class.

4.2.5 Aggregation strategies

The section level classification, while providing valuable insights into the localized content of a paper, is not the final objective of the architecture. These predictions must be mapped to a holistic decision for the entire paper. For that purpose, paper level classification is introduced, to fully simulate the final acceptance or rejection of an entire submission. Various aggregation strategies were considered to interpret the outputs of the section level model, enabling a more comprehensive analysis of the full paper evaluation pipeline.

The aggregation strategies employed translate multiple section level predictions into a single paper level prediction. Each approach offers a different implementation on how the sections contribute to the final decision of a paper. They provide a broad spectrum of decision making styles, ranging from inclusive to conservative:

- **Majority voting:** This strategy assigns to the paper the label that appears the most in the predictions of its underlying sections. It assumes that the most common output will be indicative of the whole paper. If most sections are accepted, that is the overall verdict, and vice versa.
- **Any rejected:** This method represents a more conservative approach to reviewing papers, classifying the entire work as rejected if a single one of its sections is rejected. This is not unrealistic, as a single problematic section can often be cause for rejection. It is, however, extra sensitive to noise because of this.
- **All rejected:** In contrast, this method is very lenient. A paper is classified as rejected only if all of its underlying sections are rejected. If one section is accepted, then so will be the entire submission. This method assumes that any positive content within the work can warrant acceptance.
- **Confidence weighted:** This strategy incorporates the model's prediction probabilities to weight the influence of the different sections on the final decision. Rather than simply counting the section decisions, it averages the softmax scores of the predictions across all sections and selects the label with the higher total confidence. This method incorporates not only the final decisions of the model, but also the certainty with which those predictions have been made, introducing another degree of nuance into the decision.

4.2.6 Aggregation strategy performance evaluation

This section evaluates the effectiveness of various aggregation strategies in producing accurate paper-level classifications. The aim, by comparing these methods, is to identify the approach that best captures the overall decision for a research paper based on the parts that comprise it.

Majority voting

The majority voting method yielded a strong performance, as seen in Table 4.2, achieving an accuracy of 83%. The precision for the rejected class was particularly high at 0.94, indicating that it is not overly aggressive when it comes to rejecting papers. The recall of 0.71 further supports this, suggesting this method is generally more lenient. The inverse happened for the accepted class, as is to be expected. Its recall is very high, but at the expense of precision. Still, the high individual and averaged F1-scores indicate a solid overall balance over these two metrics.

Method	Class	Precision	Recall	F1-Score	Support
Majority vote	rejected	0.94	0.71	0.81	203
	accepted	0.76	0.95	0.84	189
	Accuracy			0.83	392
	Macro average	0.85	0.83	0.83	392
	Weighted average	0.85	0.83	0.82	392

TABLE 4.2: Aggregation results using majority vote strategy

Any rejected

The any rejected strategy, whose performance can be examined in Table 4.3, adopts a far stricter approach to evaluation. As expected, the recall of this method's rejected class is extremely high, meaning that almost all rejected papers were identified. This comes at the cost of a far lower recall for the accepted class, indicating a large number of false negatives. It does, however, achieve very high precision. The overall accuracy is slightly lower, at 78%, and the F1-score for both classes also reflects this trade off. This strategy is more akin to very strict and conservative evaluation styles that are okay with rejecting borderline submissions, as long as any subpar papers do not make it through as well.

Method	Class	Precision	Recall	F1-Score	Support
Any rejected	rejected	0.71	0.97	0.82	203
	accepted	0.95	0.58	0.72	189
	Accuracy			0.78	392
	Macro average	0.83	0.78	0.77	392
	Weighted average	0.83	0.78	0.77	392

TABLE 4.3: Aggregation results using any rejected strategy

All rejected

On the opposite side of the spectrum is the all rejected strategy, which requires every section of a paper to be classified as rejected for the whole paper to be rejected. This approach leads to a complete opposite classification behaviour, as seen in Table 4.4, with near perfect recall for the accepted class, but a very poor overall performance. While the precision of the rejected papers is also quite high, at 0.97, this is insubstantial as the number of papers this method rejects is minimal. With an overall accuracy of only 57% and sub 0.5 F1-scores, this method demonstrates the worst performance across all strategies. This approach is far too lenient to be practically useful for realistic paper decision making, practically allowing all papers through unless they are uniformly bad, something that is rare in academic writing.

Method	Class	Precision	Recall	F1-Score	Support
All rejected	rejected	0.97	0.17	0.29	203
	accepted	0.53	0.99	0.69	189
	Accuracy			0.57	392
	Macro average	0.75	0.58	0.49	392
	Weighted average	0.76	0.57	0.48	392

TABLE 4.4: Aggregation results using all rejected strategy

Confidence weighted

Finally, the confidence weighted strategy performed the best overall, with an accuracy of 92%. It also achieved an F1-score of 0.92 across both classes, indicating high consistency and reliable decision making. The precision and recall was also balanced across accepted and rejected papers, suggesting the model was confident in its predictions and was able to accurately apply that confidence across papers. The above metrics are available in Table 4.5. This method highly benefits from using the section-level model's prediction probabilities, rather than only relying on the label counts. Being able to distinguished by predictions made with high confidence and ones done almost at random enhanced its performance significantly. The strong performance across all metrics indicates that the confidence weighted approach is the most effective at capturing the required information from the section-level outputs.

Method	Class	Precision	Recall	F1-Score	Support
Confidence weighted	rejected	0.98	0.87	0.92	203
	accepted	0.87	0.98	0.92	189
	Accuracy			0.92	392
	Macro average	0.93	0.92	0.92	392
	Weighted average	0.93	0.92	0.92	392

TABLE 4.5: Aggregation results using confidence weighted strategy

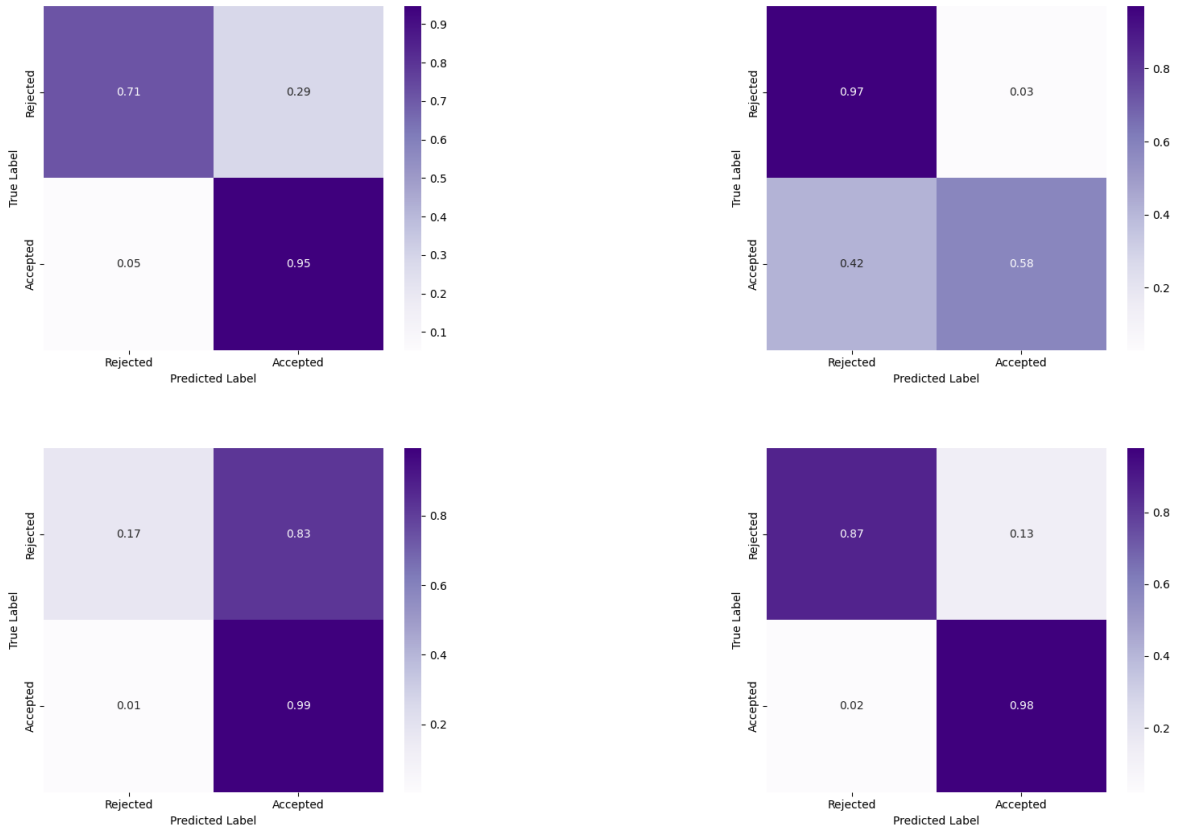


FIGURE 4.8: (Top Left) Majority voting confusion matrix (Top Right) Any rejected confusion matrix (Bottom Left) All Rejected confusion matrix (Bottom Right) Confidence weighted confusion matrix

Confusion matrices

In addition to the standard classification metrics, the performance of the different strategies is visualized using confusion matrices in Figure 4.8, which provide a more detailed account of prediction outcomes. The majority voting confusion matrix shows a balanced performance, with most mistakes occurring in the form of false negatives, and a very low rate of false positives. This confirms that the model has a tendency for being more lenient. The any rejected matrix show a high number of false positives, while capturing nearly all true negatives. This is expected due to its very conservative design. The all rejected matrix demonstrates just how rarely the model assigns the rejected label. Although nearly all accepted papers are identified, the vast majority of true rejections are missed. Finally, the confidence weighted confusion matrix shows a minimal number of misclassifications on either side, highlighting the effectiveness of the model.

Summary and comparison

To summarize, each aggregation strategy presents a different set of strengths and weaknesses. Majority voting is reliable and balanced, while remaining relatively simple. Any rejected is very aggressive in flagging problematic content, suitable for more conservative policies. All rejected, while interesting to consider, is clearly the worst performer and far too lenient to be viable in practice. Confidence weighted

stands out among the different approaches as the most effective and accurate method. Figure 4.10 provides a helpful visualization of the overall performance of the different methods.

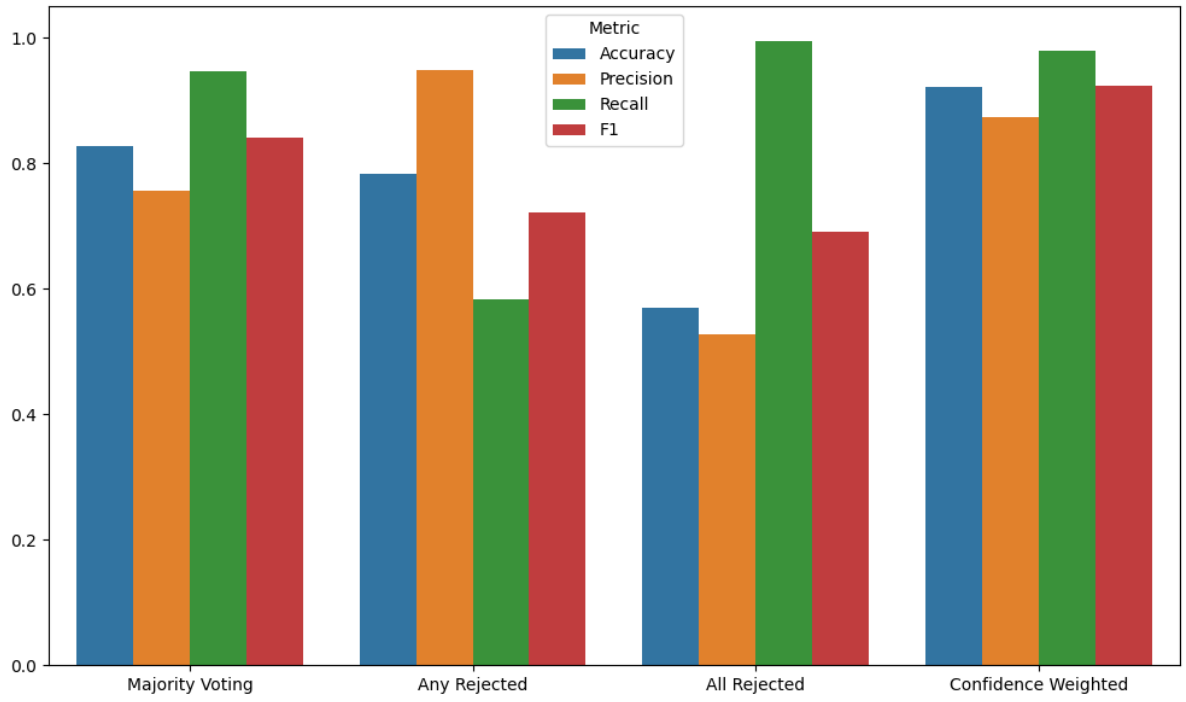


FIGURE 4.9: Aggregation strategy performance bar charts

These results demonstrate that while the section level predictions are important, the method used for the aggregation plays a crucial role in the overall quality and reliability of the paper decisions. We showed that, with the right strategy a section level classification accuracy can be propelled to over 90%, while an errant approach can cause the performance to plummet completely. It is also important to note that selecting the right aggregation method should be guided not only by absolute metrics but also by the specific goals and requirements of decision making context.

4.3 Complete paper evaluation pipeline overview

The complete flow of the proposed architecture that is designed to take in research papers and predict their acceptance status based on their content is comprised of the following sequential components:

1. **Document parsing:** The pipeline begins by ingesting a research paper in PDF format. The documents are processed through a parsing module that extracts the raw textual information.
2. **Segmentation:** The blocks of text generated by the parsing are passed through a hybrid segmentation model that classifies each segment to its role within the paper (Introduction, Abstract, etc.), dividing the document into discrete sections.
3. **Post-prediction filtering:** To address potential issues in the segmentation step, filtering mechanisms are applied that help correct inconsistencies and enforce labeling constraints.

4. **Section concatenation:** All pieces of text labeled as belonging to a specific class are added up to form complete and coherent section-level texts.
5. **Section classification:** Each reconstructed section is passed individually through a classification component which predicts whether the section belongs in an accepted or a rejected paper.
6. **Prediction aggregation:** The section-level predictions are aggregated using a defined strategy to produce a final decision for the full document: whether it is accepted or rejected.

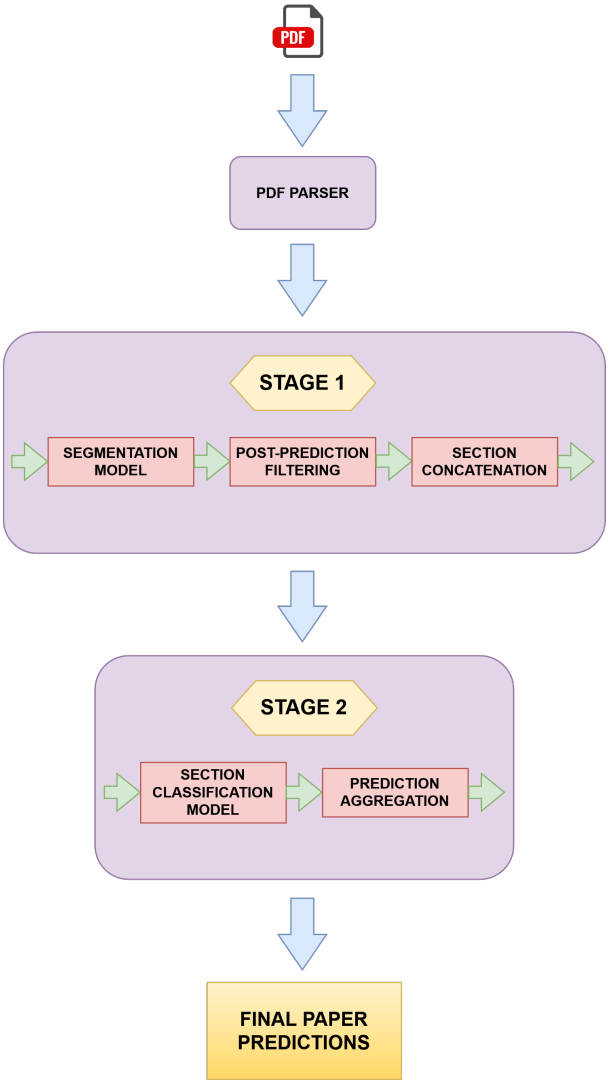


FIGURE 4.10: Paper evaluation pipeline diagram

4.4 Results

The implementation of the proposed architecture demonstrates a structured, modular approach to research paper evaluation. Beginning with parsing and segmentation of the documents to the final classification, each stage of the system was carefully designed to handle the challenge of analyzing and reviewing scientific publications.

The segmentation component, comprised of SciBERT and a sequential LSTM classifier, was trained to ascribe section labels to blocks of text. It achieved an accuracy of over 80%, a satisfying overall performance, though with a few caveats. Namely, it showed a particular weakness in identifying the section class that represents the preliminary information and related work parts of a paper. However, despite this weakness, the model proved to be more than serviceable for the purposes of the pipeline and has not been shown to negatively affect the downstream task. The subsequent post-processing that was applied also served this outcome, ironing out any obvious inconsistencies.

The section-level classifier was trained on real-world peer review decisions to learn the patterns associated with accepting or rejecting a paper. The evaluation at this level showed an accuracy of 77%, indicating that the model tasked with these localized predictions is adequately robust. This is a satisfactory, though far from perfect, performance, when we take into consideration that many of the negative or ”rejected” samples might actually be of high quality and are simply grouped in as negative due to the paper they were located in. This is an inevitable source of noise that the model was able to overcome to a certain extent.

To extend the analysis to the paper-level, multiple aggregation strategies were considered. The best performing methods were majority voting and confidence-weighted averaging, with performances of 83% accuracy for the former and an impressive 92% for the latter. It is interesting to note that both methods showed high recall for the accepted class, indicating that leniency in the evaluation leads to better classification performance.

Overall, the results confirm the viability of the system for document-level prediction, with each of its stages contributing meaningfully to the final outcome. This provides a solid foundation for future experimentation and development, whether through refining the individual components, extending the framework to a score-based rather than binary evaluation or introducing mechanisms for explainability to enhance the trustworthiness and transparency of the final decisions.

Chapter 5

Conclusions & Future Work

This chapter brings together the main outcomes of the work presented in this thesis. It provides a summary of the findings, reflects on the overall contributions while acknowledging the limitations of our approach. It also outlines possible directions for future research and experimentation.

5.1 Conclusions

The goal of this thesis was to design and implement a machine learning-based system capable of evaluating scientific research papers by emulating parts of the peer review process. To achieve this we explored the combination of document processing, natural language modeling and prediction techniques, with a particular emphasis on classification.

The thesis began by introducing key ML and NLP concepts, followed by a more focused overview of large language models and an exploration of the research process. The main part of the work involved the construction of a pipeline that processes raw PDF documents, segments them into sections that represent the different parts of a paper, and classifies them as accepted or rejected. To that end, we developed and trained a segmentation model capable of making logical splits in a paper, and a classification model for identifying whether each of the sections produced by the segmentation contributes positively or negatively to the acceptance status of the work.

With the aim of extending these prediction to the paper level, several aggregation strategies were experimented with, analyzing their ability to combine the section-level outputs into a conclusive decision for the entire paper. Through the evaluation, we found that the confidence weighted selection technique provided the most reliable and accurate results.

Our findings indicate that the proposed system is highly capable of identifying useful patterns in individual sections and accumulating them to produce reliable results. This structured, interpretable approach represents a promising framework for automating elements of the peer review process, paving the road for new research and applications.

5.2 Limitations

While the proposed system demonstrates positive results in the automated evaluation of research papers, several limitations remain that constrain its current scope and ability for generalization.

Importantly, the segmentation model is sensitive to non-standard paper layouts and documents that deviate a lot from common styling and organizational conventions can often lead to inaccurate outputs. Additionally, the model was trained on a relatively small dataset of papers due to the manual labeling requirement, which may limit its ability to generalize.

It is also worth noting that the experiments were mainly conducted with papers from computer science related fields, particularly ones that are designed to conform with conference submission rules. As a result of this, the system would most likely not be able to translate its performance to papers in other disciplines where organization, terminology and writing styles may differ significantly.

Finally, the use of BERT-based models for the purpose of classification carries the unfortunate side effect of information loss due to input size restrictions. This is not an issue for the segmentation, which handles mostly smaller blocks of text, but could significantly affect the decision classifier which receives entire sections as input. This means the model may be losing excerpts of the text that would be crucial for its final predictions.

5.3 Future Research

Building on the foundations set by this thesis, several directions for future research and enhancement can be explored. One immediate improvement over the current system would be the introduction of a score-based grading system, in place of binary classification. Rather than producing a strict accepted/rejected label, the model could output a score based on the confidence of its prediction. This would allow for more nuanced evaluation that could help distinguish borderline submissions in need of refinement from very high quality works.

Another promising idea is the integration of explainability mechanisms. Providing justifications that can be interpreted by humans for its section and paper-level predictions would increase the transparency and trustworthiness of the system, especially when serving as an assistive tool for review. In addition to the assessment, future versions could incorporate the generation of feedback in natural language. Instead of simple evaluation, the system could suggest improvements tailored to specific sections. This would enhance the model's value as a writing aid, helping authors perfect their manuscripts.

Bibliography

Ahlawat, S. (2025). *Linear Regression*, pages 3–40. Apress, Berkeley, CA.

Alake, R. (2022). A data scientist's guide to gradient descent and backpropagation algorithms. Accessed: 2024-12-09.

arXiv (2025). Monthly submissions statistics. Accessed: 2025-01-27.

Backhaus, K., Erichson, B., Gensler, S., Weiber, R., and Weiber, T. (2023). *Logistic Regression*, pages 265–352. Springer Fachmedien Wiesbaden, Wiesbaden.

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al. (2023). Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454.

Bank, D., Koenigstein, N., and Giryes, R. (2021). Autoencoders.

Bartz-Beielstein, T. (2024). *Supervised Learning: Classification and Regression*, pages 13–22. Springer Nature Singapore, Singapore.

Batane, T. (2010). Turning to turnitin to fight plagiarism among university students. *Journal of Educational Technology & Society*, 13(2):1–12.

Beltagy, I., Lo, K., and Cohan, A. (2019). Scibert: Pretrained language model for scientific text. In *EMNLP*.

Bengio, Y., Frasconi, P., and Simard, P. (1993). Problem of learning long-term dependencies in recurrent networks. In *1993 IEEE International Conference on Neural Networks*, pages 1183 – 1188 vol.3.

Bergstrom, C. T., West, J. D., and Wiseman, M. A. (2008). The eigenfactor™ metrics. *Journal of Neuroscience*, 28(45):11433–11434.

Bornmann, L. (2014). Do altmetrics point to the broader impact of research? an overview of benefits and disadvantages of altmetrics. *Journal of Informetrics*, 8(4):895–903.

Bornmann, L. and Daniel, H.-D. (2007). What do we know about the h index? *Journal of the American Society for Information Science and Technology*, 58(9):1381–1385.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

Bui, T. X. H. and Bui, V. H. (2024). Decoding scholarcy website: A study on its research summarization efficiency. In *Proceedings of the AsiaCALL International Conference*, volume 6, pages 71–80.

Clarivate Analytics (2024). Impact factor. https://clarivate.com/academia-government/essays/impact-factor/. Accessed: 2024-12-28.

dvgodoy (2021). Transformer illustration.

Efbrazil (2021). The scientific method.

Egghe, L. (2006). Theory and practise of the g-index. *Scientometrics*, 69(1):131–152.

Elicit (2023). Elicit: The ai research assistant.

Fitria, T. N. (2021). Grammarly as ai-powered english writing assistant: Students' alternative for writing english. *Metathesis: Journal of English Language, Literature, and Teaching*, 5(1):65–78.

Fricke, S. (2018). Semantic scholar. *Journal of the Medical Library Association: JMLA*, 106(1):145.

Friedl, J. (2006). *Mastering regular expressions*. " O'Reilly Media, Inc.".

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2024). Retrieval-augmented generation for large language models: A survey.

Goniwada, S. R. (2023). *Sentiment Analysis*, pages 165–184. Apress, Berkeley, CA.

Hearst, M., Dumais, S., Osman, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13:18 – 28.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2021). Measuring massive multitask language understanding.

Hicks, D., Wouters, P., Waltman, L., de Rijcke, S., and Rafols, I. (2015). The leiden manifesto for research metrics. *Nature*, 520:429–431.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.

Hu, Z., Wang, L., Lan, Y., Xu, W., Lim, E.-P., Bing, L., Xu, X., Poria, S., and Lee, R. K.-W. (2023). Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models.

Igual, L. and Seguí, S. (2024). *Basics of Natural Language Processing*, pages 195–210. Springer International Publishing, Cham.

Ioannidis, J. (2005). Why most published research findings are false. *PLoS medicine*, 2:e124.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. (2017). Quantization and training of neural networks for efficient integer-arithmetic-only inference.

Jadon, A., Patil, A., and Jadon, S. (2022). A comprehensive survey of regression based loss functions for time series forecasting.

Jain, S. M. (2022). Hugging face. In *Introduction to transformers for NLP: With the hugging face library and models to solve problems*, pages 51–67. Springer.

Jeong, C. (2024). Domain-specialized llm: Financial fine-tuning and utilization method using mistral 7b. *Journal of Intelligence and Information Systems*, 30(1):93–120.

Jo, T. (2023a). *Recurrent Neural Networks*, pages 247–275. Springer International Publishing, Cham.

Jo, T. (2023b). *Supervised Learning*, pages 29–55. Springer International Publishing, Cham.

Jo, T. (2024). *Text Summarization*, pages 271–293. Springer Nature Switzerland, Cham.

Johnson, M. (2009). How the statistical revolution changes (computational) linguistics.

Joshi, A. V. (2023). *Perceptron and Neural Networks*, pages 57–72. Springer International Publishing, Cham.

Khan, A. (2024). *Machine Learning*, pages 143–154. Springer Nature Switzerland, Cham.

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

Koroteev, M. V. (2021). Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*.

Kung, J. Y. (2023). Elicit. *The Journal of the Canadian Health Libraries Association*, 44(1):15.

Lambert, N., Castricato, L., von Werra, L., and Havrilla, A. (2022). Illustrating reinforcement learning from human feedback (rlhf). *Hugging Face Blog*. https://huggingface.co/blog/rlhf.

Le Scao, T., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. (2023). Bloom: A 176b-parameter open-access multilingual language model.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–44.

LeNail, G. (2019). Nn-svg: Publication-ready neural network architecture schematics. *Journal of Open Source Software*, 4(33):747.

Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation.

Ma, X., Fang, G., and Wang, X. (2023). Llm-pruner: On the structural pruning of large language models.

McCulloch, A., Behrend, M., and Braithwaite, F. (2022). The multiple uses of ithenticate in doctoral education: Policing malpractice or improving research writing? *Australasian Journal of Educational Technology*, 38(1):20–32.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. (2018). Mixed precision training.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*, volume 2, pages 1045–1048.

MueenAhmed, K. and Dhubaib, B. E. A. (2011). Zotero: A bibliographic assistant to researcher. *Journal of Pharmacology and Pharmacotherapeutics*, 2(4):304–305.

National Academy of Sciences, of Engineering, N. A., and of Medicine, I. (2009). *On Being a Scientist: A Guide to Responsible Conduct in Research: Third Edition*. The National Academies Press, Washington, DC.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. (2024). A comprehensive overview of large language models.

Olivieri, A. C. (2024). *Non-linearity and Artificial Neural Networks. Multi-layer Perceptron*, pages 271–288. Springer International Publishing, Cham.

OpenAI (2023). Gpt-4 technical report. Available at https://openai.com/research/gpt-4.

Pandey, N. (2023). Part of speech tagging and named entity recognition. Accessed: 2025-01-21.

Pattanayak, S. (2023). *Convolutional Neural Networks*, pages 199–291. Apress, Berkeley, CA.

Qamar, U. and Raza, M. S. (2024). *Machine Translation Using Deep Learning*, pages 449–494. Springer Nature Switzerland, Cham.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. Available at https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Rasheed, Z., Sami, M. A., Waseem, M., Kemell, K.-K., Wang, X., Nguyen, A., Systä, K., and Abrahamsson, P. (2024). Ai-powered code review with llms: Early results.

Raute, J. and Contributors (2024). Pymupdf. https://github.com/pymupdf/PyMuPDF. Version 1.26.0.

Rodríguez-Ruiz, J., Mata-Sánchez, J. I., Monroy, R., Loyola-González, O., and López-Cuevas, A. (2020). A one-class classification approach for bot detection on twitter. *Computers & Security*, 91:101715.

Runkler, T. A. (2025). *Clustering*, pages 121–143. Springer Fachmedien Wiesbaden, Wiesbaden.

Russell, S. J. and Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition.

Sarang, P. (2023). *Dimensionality Reduction*, pages 19–52. Springer International Publishing, Cham.

Singh, D., Suraksha, K., and Nirmala, S. (2021). Question answering chatbot using deep learning with nlp. In *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6.

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J. W., Kreps, S., McCain, M., Newhouse, A., Blazakis, J., McGuffie, K., and Wang, J. (2019). Release strategies and the social impacts of language models.

Tech to Words (2023). Writefull Review: AI-Powered Writing Assistance. Accessed: 2025-01-28.

The Economist (2017). The world's most valuable resource is no longer oil, but data. *The Economist, Leaders*. Accessed: 2024-12-01.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.

Tracy, S. (2010). Qualitative quality: Eight "big-tent" criteria for excellent qualitative research. *Qualitative Inquiry*, 16:837–851.

Turnitin (2016). The detection is in the details. Accessed: 2025-01-12.

University of Sheffield Library (2024). What are bibliometrics and altmetrics? https://www.sheffield.ac.uk/library/research/metrics/about. Accessed: 2024-12-15.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

Vig, J. (2019). A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Wang, W. (2025). *Unsupervised Learning Paradigm*, pages 291–324. Springer Nature Singapore, Singapore.

WHO (2020). What is dual use research of concern? https://www.who.int/news-room/questions-and-answers/item/what-is-dual-use-research-of-concern. Accessed: 2024-12-15.

Xiao, Z. (2024). *MDP: Markov Decision Process*, pages 23–80. Springer Nature Singapore, Singapore.

Yan, W. Q. (2023). *Reinforcement Learning*, pages 141–161. Springer Nature Singapore, Singapore.

Yenduri, G., M, R., G, C. S., Y, S., Srivastava, G., Maddikunta, P. K. R., G, D. R., Jhaveri, R. H., B, P., Wang, W., Vasilakos, A. V., and Gadekallu, T. R. (2023). Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions.

Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., and Chen, E. (2024). A survey on multimodal large language models. *National Science Review*, 11(12).

Yu, J., Liang, P., Fu, Y., Tahir, A., Shahin, M., Wang, C., and Cai, Y. (2024). An insight into security code review with llms: Capabilities, obstacles and influential factors.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. (2019). Hellaswag: Can a machine really finish your sentence?

Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., and Du, M. (2023). Explainability for large language models: A survey.

Zollanvari, A. (2023). *Decision Trees*, pages 187–207. Springer International Publishing, Cham.