DETECTING THE FACT CHECK-WORTINESS OF POLITICAL SPEECH

Using Bi-directional RNNs and Transformers

Primary Paper

Gencheva, Pepa, et al. *A Context-Aware Approach for Detecting Check-Worthy Claims in Political Debates*. 2017, https://doi.org/10.26615/978-954-452-049-6_037

Jacob Friedman

ITCS 5154 Spring 2025

Jacob Freidman
ITCS 5154
Spring 2025

Table of Contents

Introduction

Problem Statement

The imperative of fact-checking has become undeniable in contemporary politics. The urgency for rapid and effective fact verification was starkly highlighted during the initial term of Donald Trump, whose administration consistently demonstrated a tendency to disseminate unsubstantiated claims, both then and now. This widespread issue necessitates significant resources, with news organizations dedicating extensive man-hours and entire teams to scrutinize political discourse for factual accuracy.

Gencheva et al. (2019) raised a compelling question regarding the potential of Natural Language Processing to determine the fact-worthiness of claims by leveraging speech, semantic context, and surrounding text. This project addresses the problem of identifying claims within a document that warrant fact-checking through text and semantic analysis. Departing slightly from the original approach, this work explores the application of bi-directional RNN and Transformer models utilized to solve this task.

Motivation and Challenges

My motivation for selecting this research topic stems from the palpable political uncertainty of our era. Society is currently saturated with a pervasive influx of misinformation and fabricated narratives, frequently amplified by political figures. This necessitates the development of accessible methods to efficiently discern which claims warrant rigorous fact-checking amidst an overwhelming volume of assertions. Furthermore, my background in Political Science provided a compelling impetus to engage in a multi-disciplinary exploration of this complex problem.

The project encountered several key challenges:

- **Extensive Hyperparameter Tuning:** The necessity for conducting comprehensive grid searches across expansive hyperparameter spaces proved to be both time-intensive and demanding.
- **Computational Resource Constraints:** Limitations in available computational resources, restricted to my local machine and Google Colab, posed obstacles given the architectural complexity introduced in the design.

- **Integration of Preprocessed Data:** While utilizing the original author's data with permission, the intricate nature of their preprocessing methodology, coupled with my distinct architectural approach, presented integration challenges.
- **Addressing Data Imbalance:** The inherent complexities associated with effectively mitigating the effects of imbalanced datasets constituted a significant hurdle.

Solution Summary

This project aims to reimplement the claim detection agent presented in "A Context-Aware Approach for Detecting Worth-Checking Claims in Political Debates”. However, instead of replicating the original implementation, we will explore the efficacy of Recurrent Neural Network (RNN) and Transformer-based architectures for this task. To ensure a fair comparison, we will adhere to the paper's preprocessing methodology and utilize the same dataset to evaluate the accuracy of our models.

To optimize model performance, we will employ a rigorous hyperparameter tuning strategy involving Grid Search with Bayesian optimization, cross-validation, and early stopping. This optimization process will be conducted iteratively across three distinct phases: an initial broad search across all hyperparameters, a subsequent focused search on optimizer-related parameters, and a final refined search targeting architecture-specific parameters. The models will then be evaluated using the hyperparameters identified as optimal in each phase.

Background/Related Work

Literature Review

The proliferation of misinformation in the digital age has spurred significant research into methods for identifying and mitigating its impact, particularly within the realm of political discourse. This section reviews relevant literature addressing key aspects of this challenge, including automated fact-checking, the dynamics of information spread on social media, and techniques for analyzing narrative structures in political texts.

Gencheva et al. (2017) directly addressed the challenge of automatically identifying claims warranting fact-checking prioritization within political discourse. To this end, they constructed a novel dataset of political debates, annotating statements based on whether they had been fact-checked by nine reputable news outlets, including CNN, NPR, and

PolitiFact. The authors then developed machine learning models to predict the check-worthiness of claims. Their approach emphasized a rich contextual representation, incorporating the broader debate context, inter-speaker dynamics, and reactions from moderators and the public. Their findings demonstrated the superior performance of their context-aware models compared to a baseline system, underscoring the significance of contextual information for this task. Specifically, they employed Support Vector Machine (SVM) and deep Feed-Forward Neural Network (FNN) classifiers, trained to categorize sentences as positive if a claim within them was fact-checked by at least one source, and negative otherwise. The resulting classifier scores were then used to rank sentences by their perceived check-worthiness.

In contrast, Chen et al. (2021) investigated the presence of political bias within social media algorithms using strategically deployed neutral social bots, termed "drifters," on Twitter. Over a five-month period, these bots were programmed to follow news sources across the political spectrum. The study's findings indicated a lack of inherent political bias in the platform's news feed algorithm. Instead, the bots' experiences were significantly shaped by the political affiliations of their initial connections. The authors concluded that the observed political biases on the platform were primarily attributable to user-driven interactions and the platform's underlying mechanisms, rather than algorithmic bias. Their methodology involved deploying 15 "drifter" bots, each initially connected to a prominent news source aligned with different points on the U.S. political spectrum. These bots operated under an identical, neutral behavior model. Data on follower counts, political alignment of connections, exposure to low-credibility content, and the bots' overall political leaning were collected daily.

Shifting focus to narrative analysis, Ash et al. (2023) introduced RELATIO, a novel text analysis method designed to quantify narrative structures within textual documents. RELATIO functions by identifying coherent entity groups and mapping the explicit relationships between them as expressed in the text. To demonstrate its utility, the authors applied RELATIO to the U.S. Congressional Record to analyze the evolution of political and economic narratives over recent decades. Their analysis revealed insights into the dynamics, sentiment, polarization, and interconnectedness of narratives within political discourse. The RELATIO method leverages semantic role labeling (SRL), a computational linguistics technique, to extract agent-action-patient triplets from sentences. The resulting high-dimensional feature space is then reduced using entity clustering and embedding-based clustering to create a more manageable yet informative representation for narrative analysis. This method was subsequently applied to the U.S. Congressional Record spanning from 1994 to 2015.

Summary of Approaches

Context-Aware Claim Detection (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017)

- **Approach:** Utilizes machine learning models (SVM and FNN) trained on a dataset of political debates annotated for fact-checked claims. The models leverage a rich input representation that incorporates the context of the entire debate, interactions between speakers, and reactions from moderators and the public to predict the check-worthiness of individual claims.
- **Pros:**
    - **Leverages Rich Context:** Incorporating various contextual elements likely leads to a more nuanced understanding of claim significance.
    - **Directly Addresses Fact-Checking Prioritization:** The approach is explicitly designed to identify claims most in need of fact-checking.
    - **Utilizes Real-World Data:** Training on a dataset of actual political debates and fact-checking decisions enhances the practical applicability of the models.
    - **Demonstrated Superior Performance:** The authors showed their models outperformed a baseline, suggesting the effectiveness of their approach.
- **Cons:**
    - **Data Dependency:** The model's performance is heavily reliant on the quality and comprehensiveness of the annotated debate dataset. Creating such datasets can be time-consuming and resource-intensive.
    - **Complexity of Feature Engineering:** Defining and extracting relevant contextual features can be a complex and potentially domain-specific task.
    - **Potential for Bias in Annotations:** The fact-checking decisions of the nine reputable sources might contain inherent biases.
    - **Computational Cost:** Training deep learning models on extensive contextual data can be computationally demanding.

Social Bot Analysis of Political Bias (Chen, Pacheco, Yang, & Menczer, 2021)

- **Approach:** Employs neutral social bots ("drifters") on Twitter to observe and analyze the dynamics of information spread and the formation of political biases within the platform's ecosystem. By tracking the bots' interactions and exposure to content based on their initial connections, the study aims to understand the influence of algorithms and user behavior on political polarization.
- **Pros:**
    - **Direct Observation of Social Media Dynamics:** Provides empirical insights into how information and bias propagate on social platforms.
    - **Identifies Drivers of Bias:** Helps distinguish between algorithmic bias and user-driven factors contributing to political polarization.
    - **Scalable Data Collection:** Deploying bots allows for the collection of longitudinal data on a significant scale.
    - **Neutral Actor Perspective:** The use of programmed neutral bots offers a controlled way to study platform dynamics without inherent human biases.
- **Cons:**
    - **Limited Scope:** The findings might be specific to the Twitter platform and may not generalize to other social media environments.
    - **Ethical Considerations:** Deploying social bots, even with neutral intent, raises ethical questions about potential manipulation or misrepresentation.
    - **Complexity of Simulating Real User Behavior:** Designing bots that perfectly mimic natural human interactions can be challenging.
    - **Indirect Relevance to Claim Detection:** While it sheds light on the environment where misinformation spreads, it doesn't directly address the task of identifying specific claims for fact-checking.

Narrative Structure Analysis (Ash, Gauthier, & Widmer, 2024)

- **Approach:** Introduces RELATIO, a novel text analysis method based on semantic role labeling and entity clustering, to quantify narrative structures in text documents. By identifying entities and their relationships, the method aims to analyze the dynamics, sentiment, and interconnectedness of narratives within political discourse, as demonstrated on the U.S. Congressional Record.

- **Pros:**
    - **Novel Method for Narrative Analysis:** Offers a new approach to understanding the underlying structures and evolution of political narratives.
    - **Identifies Key Entities and Relationships:** Provides a structured way to extract and analyze the core components of narratives.
    - **Applicable to Large Text Corpora:** Demonstrated on a substantial dataset like the Congressional Record, suggesting scalability.
    - **Offers Insights into Polarization and Sentiment:** Can reveal trends and shifts in the emotional tone and divisions within political narratives.
- **Cons:**
    - **Indirect Relevance to Fact-Checking:** While understanding narrative structures might indirectly inform the context surrounding claims, it doesn't directly identify which claims are fact-worthy.
    - **Complexity of Semantic Role Labeling:** Accurate semantic role labeling can be computationally intensive and challenging, especially with complex sentence structures.
    - **Abstraction of Narrative:** The process of reducing narratives to entity groups and relations might lose some of the nuances and subtleties of the original text.
    - **Focus on Narrative, Not Factual Accuracy:** The method primarily analyzes the structure and content of narratives, not their factual correctness.

Relation to My Approach

My project directly follows Gencheva et al (2017) in aiming to automatically identify worth-checking claims in political debates using machine learning and the same dataset. However, instead of their SVM and feed-forward network, I'm exploring RNN and Transformer architectures to potentially better capture language sequences for improved accuracy.

Unlike Chen et al.'s (2021) focus on broader social media bias or Ash et al.'s (2023) narrative analysis, my work is specifically targeted at claim detection. While those approaches offer valuable context, my research directly builds upon Gencheva et al's claim identification task by investigating more advanced sequence modeling techniques.

Methods

My Approach

This research adopts the foundational logic presented in "A Context-Aware Approach for Detecting Worth-Checking Claims in Political Debates" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017). To ensure a robust and direct comparison of results, the identical pre-processing steps employed in their study were meticulously replicated, utilizing their publicly available pre-processing code with appropriate attribution as requested in their code repository. This allowed for the precise recreation of their testing dataset, providing a consistent evaluation benchmark.

The primary divergence in this approach lies in the architectural design of the claim detection models. While Gencheva et al. (2017) utilized a Support Vector Machine (SVM) layer followed by a feed-forward neural network, this project explores the capabilities of sequence-based models. Specifically, two distinct architectures were implemented: one leveraging a Recurrent Neural Network (RNN) and the other employing a Transformer-based encoder.

Despite this fundamental change in the core sequential processing layers, several key architectural parameters were intentionally maintained. These include the number of subsequent linear layers, the number of output units, the Leaky ReLU activation function applied to intermediate layers, and the Sigmoid activation function used for the final output. It was important to maintain this core architecture for comparison purposes.

Furthermore, the task-specific branching architecture, designed to handle both single-source and combined-source claim evaluation, was preserved. The crucial modification involved replacing the initial SVM layer with a Long Short-Term Memory (LSTM) layer in the RNN-based architecture and a Transformer encoder layer in the Transformer-based architecture, respectively, to better capture the sequential dependencies inherent in political discourse.

To further investigate the impact of task weighting, the entire modeling and evaluation process was conducted twice: once with equal weighting assigned to the single-source and combined-source tasks, and a second time with a higher weight assigned to the single-source task, reflecting its potentially greater importance or distinct characteristics.

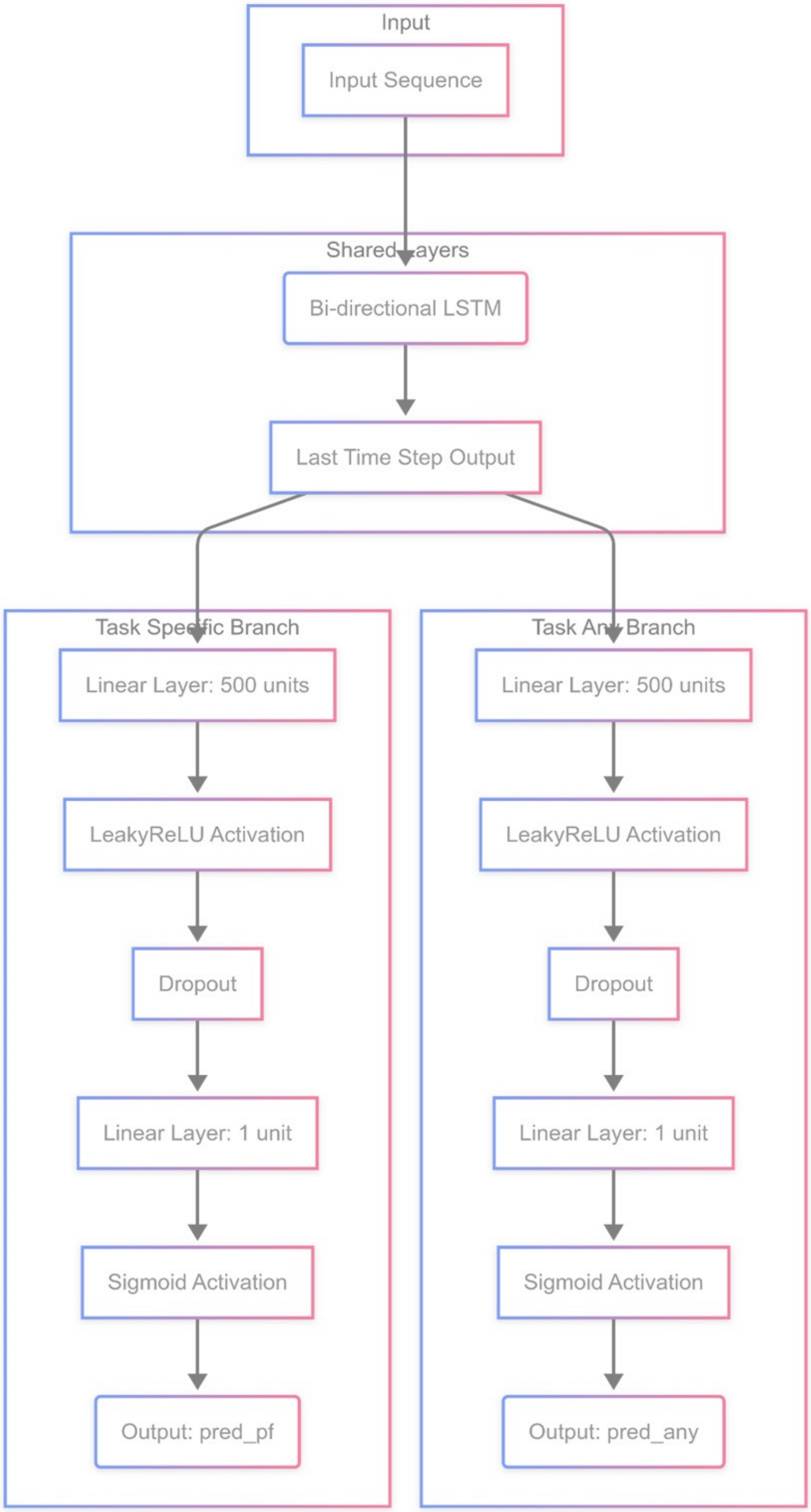
The implementation of the loss function also involved a comparative analysis of several multi-task learning strategies. The base loss function employed was binary cross-entropy, a standard choice for binary classification tasks. We take a combined average of each task score to calculate the total loss across tasks. To effectively address the multi-task nature of the problem and explore different balancing techniques, the following custom loss function implementations were developed and evaluated:

- **multitask_loss(preds, targets)**: This function calculates the overall loss by simply summing the binary cross-entropy loss computed for each individual task. This serves as a baseline for multi-task learning.
- **multitask_loss_auto_weighted_bce(preds, targets)**: This extends the basic multitask_loss by automatically assigning weights to the binary cross-entropy loss of each task based on the observed class distribution within that task. This dynamic weighting aims to mitigate the challenges posed by potential class imbalance in the different tasks.
- **multitask_loss_dynamic_weighted(preds, targets)**: Similar to the auto-weighted approach, this function dynamically adjusts the weights applied to each task's loss during the training process. The weight updates are informed by the magnitude of the loss observed for each task in the preceding training step, allowing the model to focus more on tasks with higher error.
- **multitask_loss_loss_proportional(preds, targets, epsilon=1e-8)**: This loss function implements a strategy where the weight assigned to each task's loss is inversely proportional to the task's current loss. The inclusion of a small epsilon value prevents division by zero and ensures numerical stability. This approach aims to balance the contribution of each task to the overall gradient by giving more emphasis to tasks with higher losses.

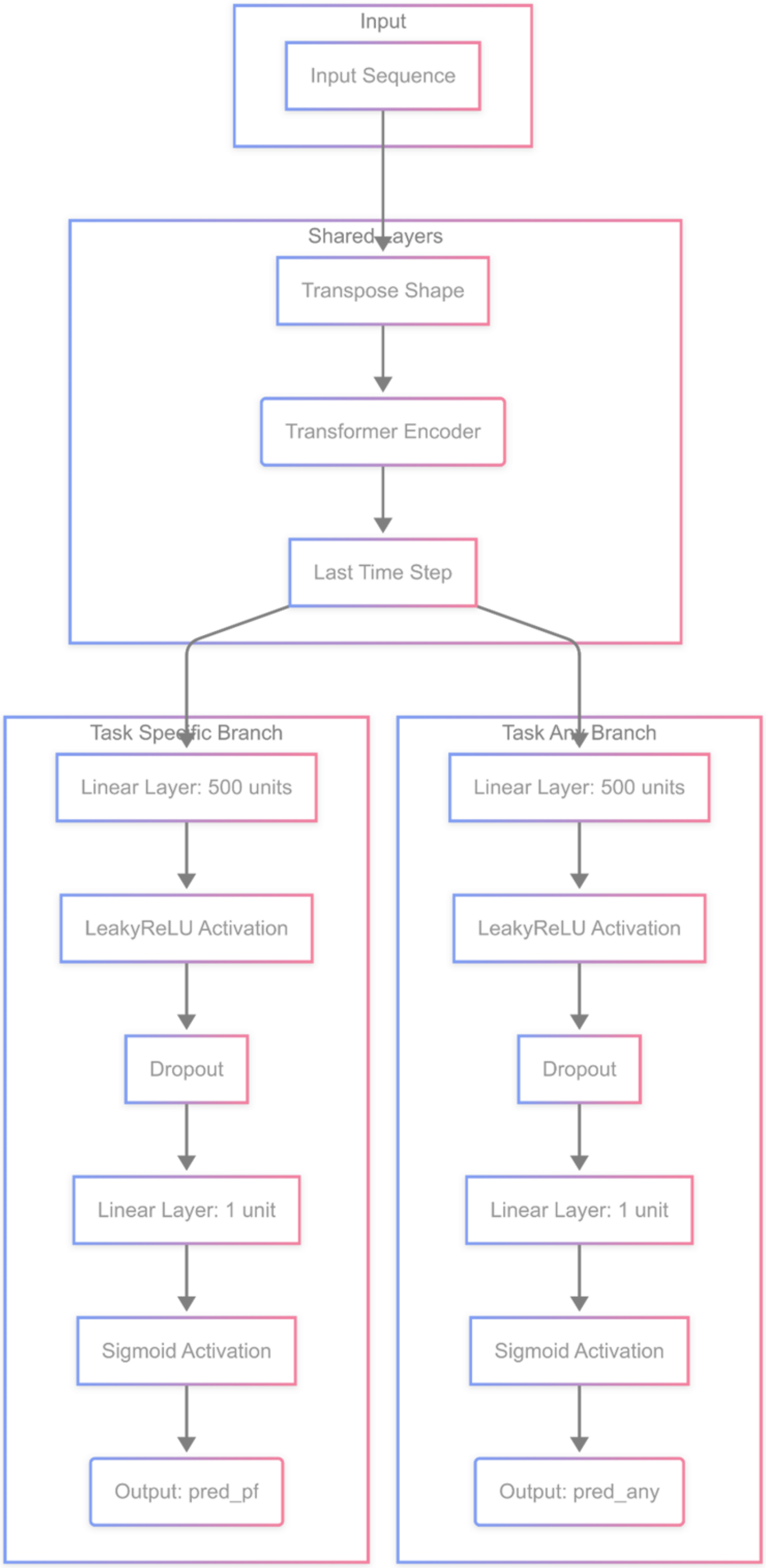
By systematically exploring these architectural variations and loss function strategies, this research aims to provide a comprehensive evaluation of the effectiveness of RNN and Transformer-based models for the task of detecting worth-checking claims in political debates, while maintaining a direct point of comparison with the original work of Gencheva et al. (2017).

Architecture Diagrams

MultiTaskRNN Architecture



MultiTaskTransformer Architecture



Data Design: The CW-USPD-2016 Dataset

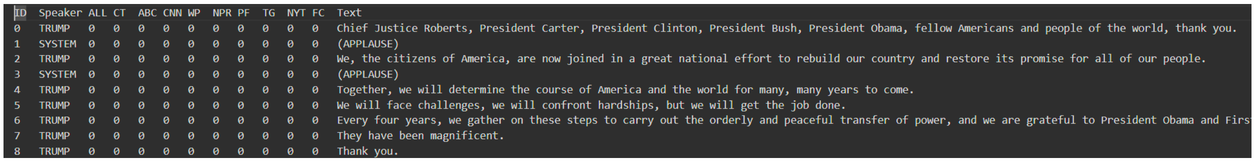
Gencheva et al. (2017) created a novel dataset specifically for identifying check-worthy claims within the context of political debates, named CW-USPD-2016 (check-worthiness in the US presidential debates 2016). The foundation of this dataset comprises transcripts from four significant political events in the 2016 US election cycle: "one vice-presidential and three presidential debates" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017).

To annotate these transcripts, the researchers gathered publicly available manual analyses from nine distinct, reputable fact-checking sources. These sources included well-known organizations like "CNN, NPR, and PolitiFact" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017), among others such as ABC News, Chicago Tribune, FactCheck.org, The Guardian, The New York Times, and The Washington Post. The authors explain that for each debate, they "used the publicly-available manual analysis about it from nine reputable fact-checking sources" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017). This analysis wasn't limited to factuality statements but could encompass "any free text that journalists decided to add, e.g., links to biographies or behavioral analysis of the opponents and moderators" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017).

The raw analyses from these sources were then processed into a usable format for machine learning. The authors "converted this to binary annotation about whether a particular sentence was annotated for factuality by a given source" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017). Specific rules were applied during this conversion: "Whenever one or more annotations were about part of a sentence, we selected the entire sentence, and when an annotation spanned over multiple sentences, we selected each of them" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017). This process resulted in a dataset containing "four debates, with a total of 5,415 sentences" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017).

An important characteristic of this dataset is the relatively low agreement between the different fact-checking sources regarding which sentences were check-worthy. Gencheva et al. (2017) note, "only one sentence was selected by all nine sources, 57 sentences by at least five, 197 by at least three, 388 by at least two, and 880 by at least one". They attribute this low agreement to the fact that "the different media aimed at annotating sentences according to their own editorial line, rather than trying to be exhaustive in any way" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017). This inherent variability highlights the challenge of the task and informed the authors' decision to model the problem as a ranking task rather than absolute prediction. They explicitly state that they "focus on a ranking task rather than on absolute predictions" (Gencheva, Koychev, Màrquez, Barrón-Cedeño, & Nakov, 2017). The dataset was designed to capture claims within their full conversational context, unlike previous work that often looked at sentences in isolation.

Here is an example of one of their annotated files for the 2016 Trump Inauguration Address transcript:



```
ID  Speaker ALL CT  ABC CNN WP  NPR PF  TG  NYT FC  Text
0   TRUMP   0   0   0   0   0   0   0   0   0   0   Chief Justice Roberts, President Carter, President Clinton, President Bush, President Obama, fellow Americans and people of the world, thank you.
1   SYSTEM  0   0   0   0   0   0   0   0   0   0   (APPLAUSE)
2   TRUMP   0   0   0   0   0   0   0   0   0   0   We, the citizens of America, are now joined in a great national effort to rebuild our country and restore its promise for all of our people.
3   SYSTEM  0   0   0   0   0   0   0   0   0   0   (APPLAUSE)
4   TRUMP   0   0   0   0   0   0   0   0   0   0   Together, we will determine the course of America and the world for many, many years to come.
5   TRUMP   0   0   0   0   0   0   0   0   0   0   We will face challenges, we will confront hardships, but we will get the job done.
6   TRUMP   0   0   0   0   0   0   0   0   0   0   Every four years, we gather on these steps to carry out the orderly and peaceful transfer of power, and we are grateful to President Obama and Firs
7   TRUMP   0   0   0   0   0   0   0   0   0   0   They have been magnificent.
8   TRUMP   0   0   0   0   0   0   0   0   0   0   Thank you.
```

Experimentation

Experimental Steps

Data Pre-processing

The data underwent a pre-processing stage that closely mirrored the methodology established by the original authors. Recognizing the complexity of their feature engineering process and ensuring a direct and fair comparison of results, their pre-processing pipeline was adopted, with minor refactoring for contemporary code standards. This decision mitigated the risk of introducing discrepancies arising from a reimplementation of intricate features.

The authors' pre-processing involved several key steps:

First, the raw debate transcripts were loaded and organized into training and validation datasets. This initial step focused on acquiring and partitioning the textual data.

Second, a crucial feature extraction pipeline was applied to convert the textual content into a numerical format suitable for machine learning models. This pipeline, leveraging pre-computed features for efficiency, encompassed the extraction of contextual information surrounding each claim, part-of-speech tags to capture syntactic elements, and stylistic features reflecting the writing style. This transformation yielded a numerical matrix representation of the debate data.

Finally, corresponding labels indicating the check-worthiness of claims were extracted for various evaluation scenarios. These labels represented assessments from different fact-checking sources, including a general binary label for overall check-worthiness and specific labels derived from sources like PolitiFact and the Washington Post. This step prepared the target variables for model training and evaluation against different fact-checking standards.

Model Implementation

Two distinct neural network architectures were implemented for claim detection based on the original architecture of Gencheva et al. (2017): a Recurrent Neural Network (RNN)-based model and a Transformer-based model.

The RNN model, MultiTaskRNN, was designed for sequence processing. It incorporated a shared Long Short-Term Memory (LSTM) layer to capture temporal dependencies within the input text. Following the shared LSTM, task-specific branches, consisting of linear layers, Leaky ReLU activation, dropout, and a final Sigmoid activation, were employed to generate predictions for the different fact-checking tasks.

The Transformer model, MultiTaskTransformer, utilized a shared Transformer Encoder to process the input sequences, leveraging self-attention mechanisms to understand contextual relationships. Similar to the RNN model, task-specific branches with linear layers, Leaky ReLU activation, dropout, and a final Sigmoid activation were used to produce predictions for each task.

Model Training and Evaluation

A systematic hyperparameter optimization was conducted for both the RNN and Transformer models, employing Bayesian Optimization in conjunction with k-fold cross-validation. This strategy facilitated an efficient exploration of the hyperparameter space to pinpoint configurations that minimized validation loss and exhibited robust generalization across different data partitions. To mitigate overfitting and optimize computational efficiency, an early stopping mechanism was integrated into the training procedure.

For the RNN model, the hyperparameter search aimed to identify optimal settings for key tuning parameters. The model was subsequently trained on the training data using the identified best hyperparameter configuration. Following training, its performance was evaluated on the validation set, with relevant performance metrics recorded and visualizations generated to facilitate a comprehensive analysis of the results.

Similarly, the Transformer model underwent a thorough hyperparameter search to determine its optimal configuration. The model was then trained using these optimal parameters, and its performance was assessed on the validation set, with corresponding metrics and visualizations produced for detailed analysis.

The training of both models utilized the binary cross-entropy loss function as a foundation. To effectively address the multi-task learning nature of the problem and potential imbalances in class distributions, several variations of the loss function were explored. These included a basic summation of individual task losses, an approach that automatically weighted task losses based on class prevalence, a dynamic weighting scheme that adjusted task weights during training based on recent loss values, and a proportional weighting strategy where task weights were inversely related to their respective losses.

The culmination of the training and optimization process involved evaluating the final trained models on the held-out validation set. The results of this evaluation were then visualized to provide insights into the models' effectiveness in predicting the worthiness of claims for fact-checking across different fact-checking sources.

This entire hyperparameter optimization and training cycle was executed iteratively across three distinct phases. The initial phase involved a broad search encompassing all tunable hyperparameters. The subsequent phase focused specifically on optimizing parameters related to the training optimizer. Finally, the third phase concentrated on fine-tuning hyperparameters governing the model's architecture. The optimal hyperparameter configuration identified after the conclusion of the third phase was then utilized for the final model training and subsequent in-depth analysis of the results.

Experimental Treatments

This study employed a series of experimental treatments to evaluate the performance of the implemented Recurrent Neural Network (RNN) and Transformer models under various conditions. These treatments were designed to build upon a baseline established with author-suggested hyperparameters and then explore the impact of task weighting strategies.

1. Baseline Treatment: This initial treatment aimed to replicate a standard experimental setup, utilizing hyperparameter settings suggested by Gencheva et al. (2017) for several key configurations. The performance of both model architectures was assessed using the average validation loss as the primary evaluation metric.

1. **RNN with Average Validation Loss**: The RNN model was trained and evaluated using the average validation loss.
2. **Transformer with Average Validation Loss**: The Transformer model was trained and evaluated using the average validation loss.

2. Task Un-Weighted Treatment: This treatment investigated the models' performance when treating the single-source and combined-source prediction tasks with equal importance during training.

1. **RNN with Average Validation Loss**: The RNN model was trained and evaluated using the average validation loss.
2. **RNN with Combined Score Validations**: The RNN model was evaluated using both the F1 score and average loss on a combined score derived from the individual task predictions.
3. **Transformer with Average Validation Loss**: The Transformer model was trained and evaluated using the average validation loss.
4. **Transformer with Combined Score Validations**: The Transformer model was evaluated using both the F1 score and average loss on a combined score derived from the individual task predictions.

3. Task Weighted Treatment: This treatment explored the effect of assigning a higher weight to the single-source prediction task during training, hypothesizing that this task might hold more salient information or present a different level of difficulty.

1. **RNN with Average Validation Loss**: The RNN model was trained with a higher weight on the single-source task and evaluated using the average validation loss.
2. **RNN with Combined Score Validations**: The RNN model, trained with weighted tasks, was evaluated using both the F1 score and average loss on a combined score derived from the individual task predictions.
3. **Transformer with Average Validation Loss**: The Transformer model was trained with a higher weight on the single-source task and evaluated using the average validation loss.
4. **Transformer with Combined Score Validations**: The Transformer model, trained with weighted tasks, was evaluated using both the F1 score and average loss on a combined score derived from the individual task predictions.

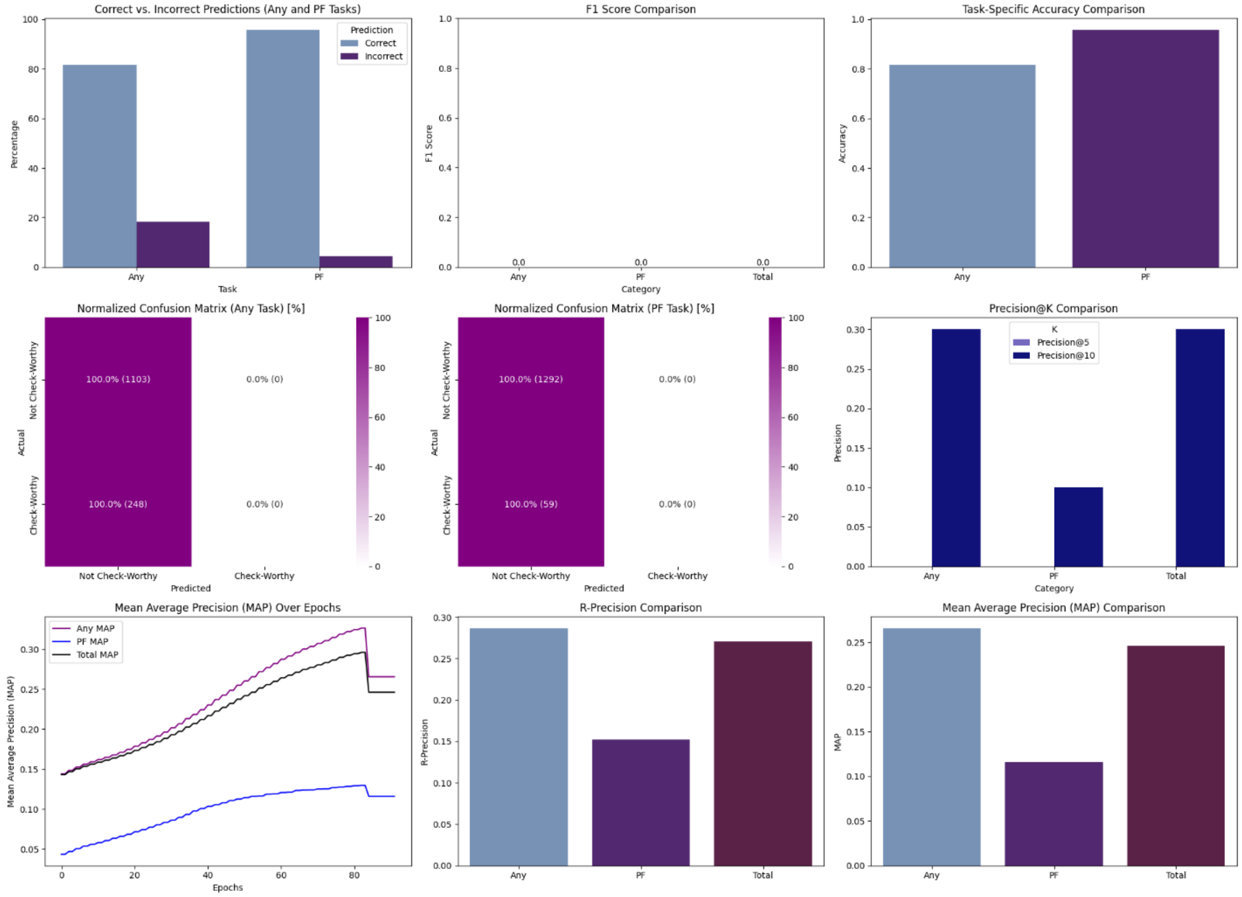
Through these distinct experimental treatments, this study aimed to systematically analyze the impact of model architecture and task weighting strategies on the performance of claim detection models. The inclusion of evaluations based on a combined score provides a more holistic understanding of the models' predictive capabilities across different evaluation perspectives.

You can view my GitHub for more information including templates and a link to my repository as a Zip file.

Experimental Results

Baseline Treatment

MultiTaskRNN Results



The results for the Baseline RNN treatment, evaluated using average validation loss, provide an initial assessment of the model's capabilities without specific task weighting.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart indicates a significant imbalance in the predictions. For the "Any" task, a large majority (around 80%) of the instances were predicted correctly, while a smaller proportion (around 20%) were incorrect. The "PF" task shows an even stronger trend, with a very high percentage (over 90%) predicted correctly and a small fraction incorrectly. This suggests the model, in its baseline configuration, exhibits a tendency towards the majority class for both prediction targets.

F1 Score Comparison: The F1 scores for the "Any" (approximately 0.0) and "Total" (approximately 0.0) categories are extremely low. This indicates a significant deficiency in the model's ability to balance precision and recall for identifying worth-checking claims. An F1 score close to zero suggests the model is likely either failing to identify positive instances or generating a high number of false positives.

Task-Specific Accuracy Comparison: The accuracy for the "Any" task is around 0.8, while the accuracy for the "PF" task is very high, close to 1.0. The high accuracy on the "PF" task, despite the low F1 score, likely arises from the class imbalance, where the model might be achieving high accuracy by predominantly predicting the majority class.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task reveals that when the actual label was "Not Check-Worthy," the model correctly predicted it 100% of the time. However, when the actual label was "Check-Worthy," the model never correctly identified it. This highlights the model's bias towards predicting "Not Check-Worthy" for the "Any" task.

Normalized Confusion Matrix (PF Task): The confusion matrix for the "PF" task shows a perfect 100% prediction for "Not Check-Worthy". While "Check-Worthy" instances showed a very dismal 0%. However, this perfect accuracy from “Not Check-Worthy” is likely misleading, as the model could be simply predicting the majority class for all instances.

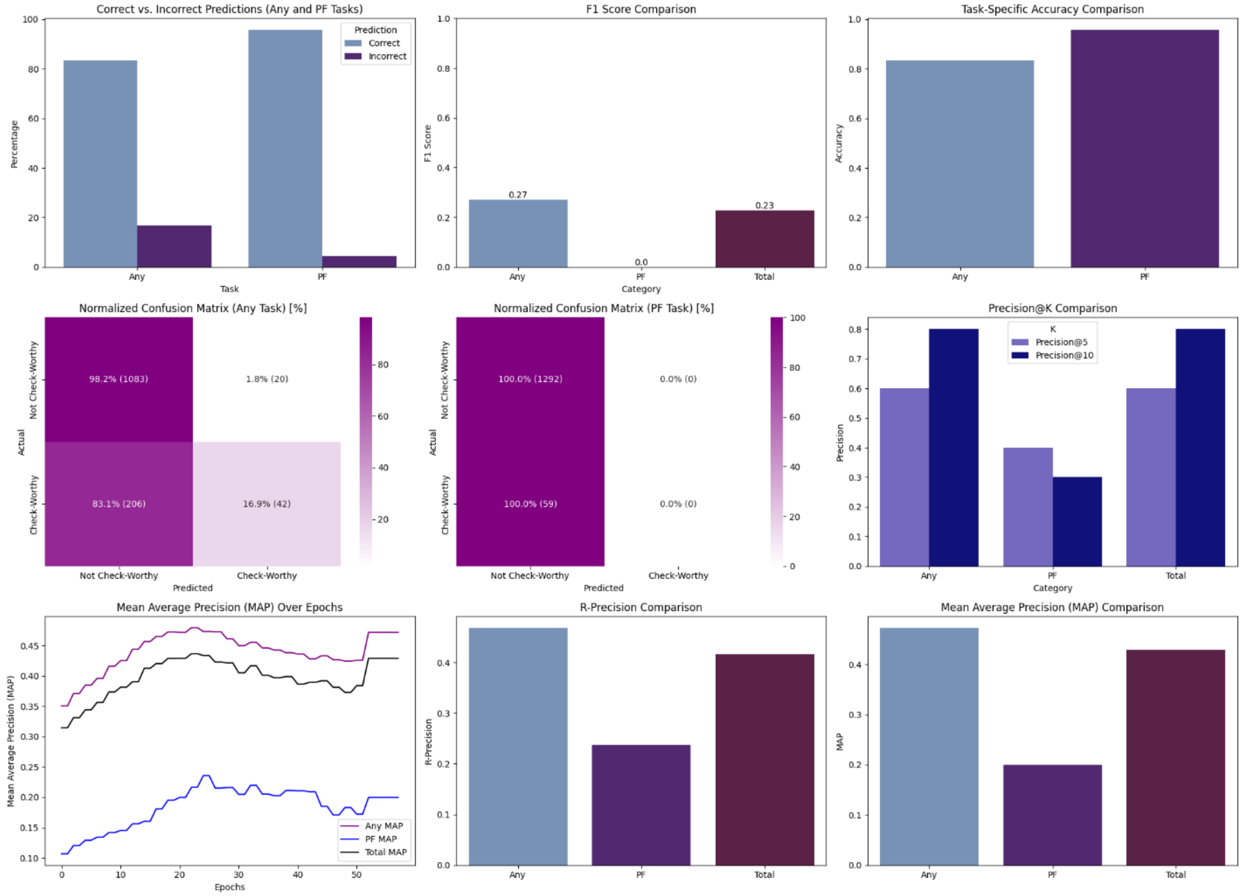
Precision @ K Comparison: The precision at K (for K=5 and K=10) is generally low across the "Any" and "Total" tasks, hovering around 0.3 for "Any" and "Total”. While lying around 0.1 for the “PF” task. This indicates that when the model provides its top 5 or top 10 most likely worth-checking claims, the proportion of truly worth-checking claims within that set is not high.

Mean Average Precision (MAP) Over Epochs: The MAP score for the "Any" task shows some fluctuation over the epochs but generally remains below 0.35. The MAP score for the "PF" task starts very low (near 0.0) and shows minimal improvement. The "Total" MAP score also remains relatively low throughout training. This suggests that the model is not effectively learning to rank the worth-checking claims such that the truly positive instances appear higher in the ranking.

R-Precision Comparison: The R-Precision is low for the "Any" category (around 0.29) and also low for the "Total" category (around 0.27), further indicating poor retrieval performance of truly worth-checking claims.

Mean Average Precision (MAP) Comparison: The final MAP comparison shows very low scores for the "Any" (around 0.25) and "PF" (around 0.1) tasks, with a slightly higher but still low score for the "Total" category (around 0.24).

MultiTaskTransformer Results



The results for the Baseline Transformer treatment, evaluated using average validation loss, offer an initial perspective on the Transformer architecture's performance without specific task weighting.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart shows that for the "Any" task, a large majority (around 80%) of predictions were correct, with a smaller portion (around 20%) being incorrect. The "PF" task exhibits even higher correct predictions (over 90%) compared to incorrect predictions (under 10%). This suggests the model tends to correctly classify instances, particularly for the "PF" task.

F1 Score Comparison: The F1 score for the "Any" task is approximately 0.27, and for the "Total" task, it's around 0.23. These relatively low scores indicate a poor balance between precision and recall, suggesting the model struggles to accurately identify positive (worth-checking) claims without a significant number of false positives or false negatives.

Task-Specific Accuracy Comparison: The accuracy for the "Any" task is notably high, around 0.8. The accuracy for the "PF" task is even higher, approaching 1.0. The high accuracy, especially on the "PF" task, despite the low F1 score, likely reflects a class imbalance where the model might be achieving high accuracy by predominantly predicting the majority class.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task reveals that when the actual label was "Not Check-Worthy," the model correctly predicted it 98.2% of the time. However, when the actual label was "Check-Worthy," the model only correctly identified it 16.9% of the time, misclassifying 83.1% as "Not Check-Worthy". This indicates a bias towards predicting the "Not Check-Worthy" class for the "Any" task.

Normalized Confusion Matrix (PF Task): The confusion matrix for the "PF" task shows a perfect 100% prediction for "Not Check-Worthy". While "Check-Worthy" instances showed a disappointing 0%. However, this perfect accuracy from “Not Check-Worthy” is likely misleading, as the model could be simply predicting the majority class for all instances.

Precision @ K Comparison: The Precision @ K (for K=5 and K=10) reveals a moderate level of accuracy for the "Any" category (approximately 0.6 and 0.8, respectively) and a similar, though still modest, accuracy for the "Total" category (around 0.6 and 0.8, respectively). While the “PF” task shows poor performance in comparison (around 0.4 and 0.3, respectively). This suggests that even within the model's top 5 or 10 predictions for worth-checking claims, the proportion of genuinely worth-checking claims remains limited.

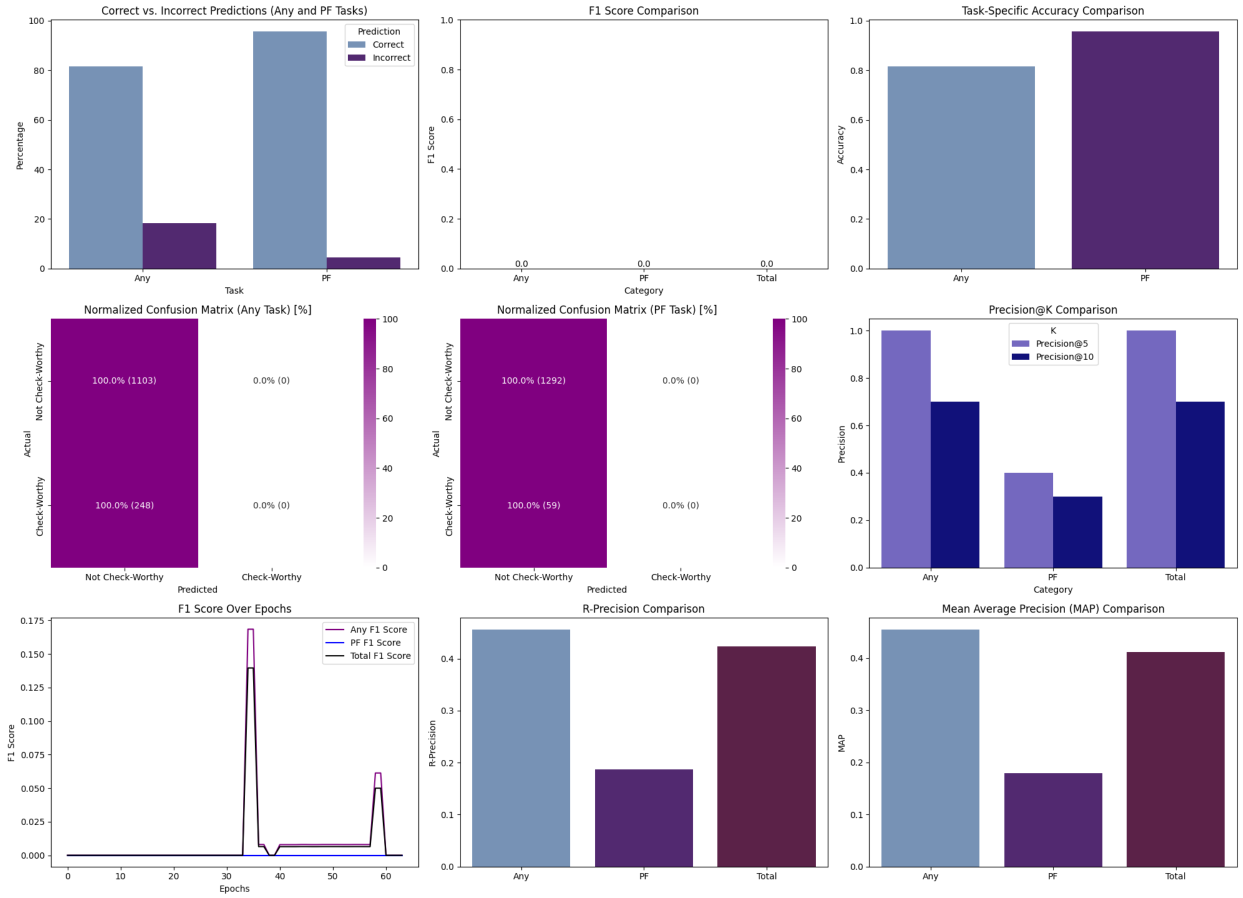
Mean Average Precision (MAP) Over Epochs: The MAP score for the "Any" task hovers below 0.45, indicating a limited ability to rank worth-checking claims effectively. The "PF" task shows minimal improvement from a near-zero MAP to just 0.2, demonstrating a significant issue in prioritizing positive instances. The "Total" MAP also remains low, staying under 0.4 throughout training, further suggesting the model's difficulty in placing truly worth-checking claims higher in the ranking.

R-Precision Comparison: With R-Precision scores of around 0.44 ("Any"), 0.4 ("Total"), and a very low 0.22 ("PF"), the model demonstrates poor retrieval of truly worth-checking claims at the rank of relevant documents.

Mean Average Precision (MAP) Comparison: The final MAP scores are low across the board: around 0.44 for "Any," 0.2 for "PF," and a slightly higher but still low 0.42 for "Total." This highlights the model's limited success in effectively ranking worth-checking claims.

Task Un-Weighted Treatment

MultiTaskRNN with Average Loss Validation Results



This section examines the performance of the RNN model under the Task Un-Weighted treatment, where the losses for the "Any" and "PF" tasks were given equal importance during training, and the evaluation is based on the average validation loss.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart shows a similar pattern to the Baseline RNN. For the "Any" task, a large majority of instances are predicted correctly, with a smaller portion being incorrect. The "PF" task also exhibits a high rate of correct predictions, but the imbalance between correct and incorrect predictions appears similar to the baseline.

F1 Score Comparison: The F1 score for the "Any" task remains low, around 0.0. The F1 score for the "Total" task is also very low, near 0.0. This indicates that treating the tasks with equal weight, in this configuration, has had no impact on the model's ability to achieve a balance between precision and recall for identifying worth-checking claims.

Task-Specific Accuracy Comparison: The accuracy for the "Any" task is notably high, around 0.8. The accuracy for the "PF" task is even higher, approaching 1.0. The high accuracy, especially on the "PF" task, despite the low F1 score, likely reflects a class imbalance where the model might be achieving high accuracy by predominantly predicting the majority class.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task shows that when the actual label was "Not Check-Worthy," the model correctly predicted it 100.0% of the time. However, when the actual label was "Check-Worthy," the model incorrectly classified all instances (0.0% correct).

Normalized Confusion Matrix (PF Task): Similar to the baseline, the "PF" task confusion matrix shows 100.0% correct and incorrect predictions for both classes, respectively. Again, given the likely class imbalance, this perfect accuracy is likely due to the model consistently predicting the majority "Not Check-Worthy" class.

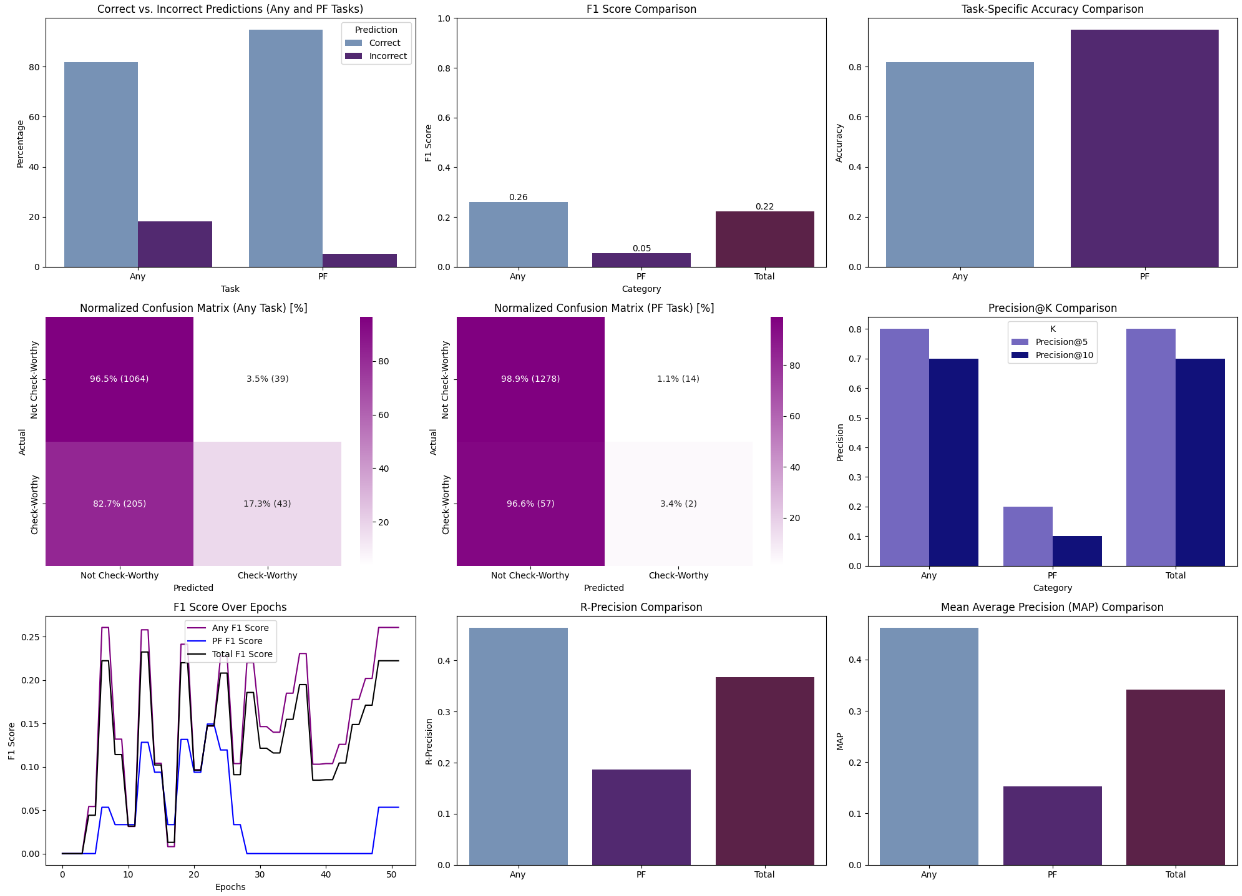
Precision @ K Comparison: The Precision @ K (for K=5 and K=10) for the "Any" and “Total” categories is roughly 1.0 and 0.6, respectively. The Precision @ K for the "PF" category is also very low. This reinforces the finding that the model, under this un-weighted treatment, is ineffective at retrieving relevant worth-checking claims within its top predictions.

F1 Score Over Epochs: The F1 score for the "Any" task remains at 0.0 throughout most of the training epochs, with a slight and late increase. The F1 score for the "Any" task starts low and shows a minor increase towards the end. The "Total" F1 score mirrors the "Any" task, staying very low for the majority of training. This suggests that the un-weighted approach does not facilitate effective learning for balanced prediction.

R-Precision Comparison: The R-Precision for the "Any" task is 0.43, and it is also very low for the "Total" task at 0.41. With a very poor 0.2 for the “PF” task. This confirms the model's inability to effectively retrieve truly worth-checking claims when the tasks are unweighted.

Mean Average Precision (MAP) Comparison: The final MAP comparison shows a MAP score of 0.4 for the "Any" and “Total” task, with a 0.2 for the “PF” task. The MAP score for the "PF" task remains very low, and the “Any” and "Total" MAP score is also low.

MultiTaskRNN with Combined Score Validations Results



This section analyzes the performance of the RNN model under the Task Un-Weighted treatment, where the losses for the "Any" and "PF" tasks were equally weighted during training, and the evaluation is based on a combined score utilizing both F1 score and average loss on the validation set.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart shows a similar trend to the previous RNN treatments. For the "Any" task, a large majority of instances are predicted correctly, with a smaller portion being incorrect. The "PF" task also exhibits a high rate of correct predictions, with a very small percentage of incorrect predictions.

F1 Score Comparison: The F1 score for the "Any" task has improved to approximately 0.26 compared to the 0.0 observed with average loss validation in the un-weighted setting. The F1 score for the "Total" task is around 0.22. This suggests that evaluating based on a combined score has positively influenced the model's ability to balance precision and recall for the "Any" task. But the “PF” task still exhibits a very poor 0.05 score. The F1 scores still indicate a need for further improvement.

Task-Specific Accuracy Comparison: The accuracy for the "Any" task is around 0.8. The accuracy for the "PF" task remains very high, approximately 0.97. The improvement in "Any" task accuracy likely contributes to the better F1 score.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task shows that when the actual label was "Not Check-Worthy," the model correctly predicted it 96.5% of the time. When the actual label was "Check-Worthy," the model correctly identified it 17.3% of the time, with 82.7% being misclassified as "Not Check-Worthy". This represents a significant improvement in identifying "Check-Worthy" claims compared to the average loss validation setting.

Normalized Confusion Matrix (PF Task): The confusion matrix for the "PF" task shows a high correct prediction rate for both "Not Check-Worthy" (98.9%) and a slightly improved rate for "Check-Worthy" (3.4%) instances. While still showing some imbalance in the number of instances, the model demonstrates a better ability to identify both classes compared to baseline treatments.

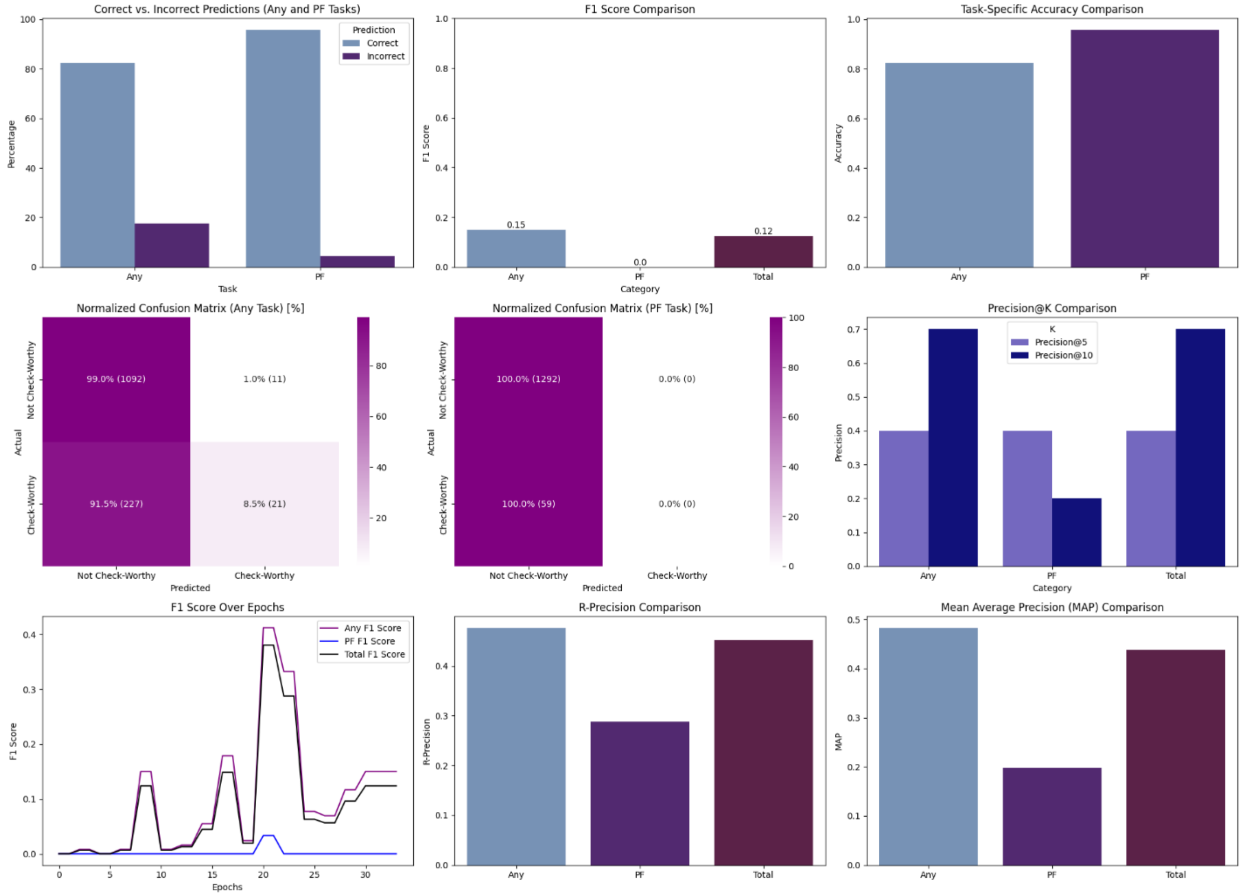
Precision @ K Comparison: The Precision @ K (for K=5 and K=10) for the "Any" task shows a substantial improvement compared to the average loss validation, reaching around 0.8 and 0.7, respectively. The Precision @ K for the "Total" category is also around 0.8 and 0.7, respectively. This indicates that when the model provides its top predicted worth-checking claims, a larger proportion of them are actually worth checking, while it still struggles with “PF” task prediction at 0.2 and 0.1, respectively.

F1 Score Over Epochs: The F1 score for the "Any" task shows a more dynamic learning curve, with increases and plateaus throughout training, reaching a higher final value than in the average loss validation setting. The "Total" F1 score also shows improvement over epochs. This suggests that the combined score validation guides the model towards a better balance of precision and recall during training.

R-Precision Comparison: The R-Precision for the "Any" task has improved significantly to around 0.44, and the "Total" task R-Precision is around 0.37. This confirms that the model is now better at retrieving truly worth-checking claims within the top ranks.

Mean Average Precision (MAP) Comparison: The final MAP comparison reveals a consistent performance in the "Any" task's MAP score when compared to the average loss validation setting. Similarly, the MAP scores for the "Total" and "PF" tasks show comparable results. This suggests that the model's ability to rank worth-checking claims higher in its predictions remains consistent across these settings.

MultiTaskTransformer with Average Loss Validation Results



This section examines the performance of the Transformer model under the Task Un-Weighted treatment, where the losses for the "Any" and "PF" tasks were given equal importance during training, and the evaluation is based on the average validation loss.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart indicates a strong imbalance in predictions. For the "Any" task, a large majority are predicted correctly, with a smaller portion incorrect. The "PF" task shows a near-perfect rate of correct predictions.

F1 Score Comparison: The F1 score for the "Any" task is approximately 0.15, a slight improvement over the Baseline Transformer. The F1 score for the "Total" category is around 0.12, also a modest increase. These scores still indicate a poor balance between precision and recall.

Task-Specific Accuracy Comparison: The accuracy for the "Any" task is around 0.8, similar to the baseline. The accuracy for the "PF" task remains very high, approximately 0.99.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task shows that when the actual label was "Not Check-Worthy," the model correctly predicted it 99.0% of the time. However, when the actual label was "Check-Worthy," the model only correctly identified it 8.5% of the time, with 91.5% being misclassified as "Not Check-Worthy." This indicates a strong bias towards predicting "Not Check-Worthy" for the "Any" task.

Normalized Confusion Matrix (PF Task): The confusion matrix for the "PF" task shows 100.0% correct predictions for "Not Check-Worthy" instances and 0.0% correct for "Check-Worthy" instances. Given the likely class imbalance, this suggests the model is consistently predicting the majority "Not Check-Worthy" class for the "PF" task in this un-weighted setting as well.

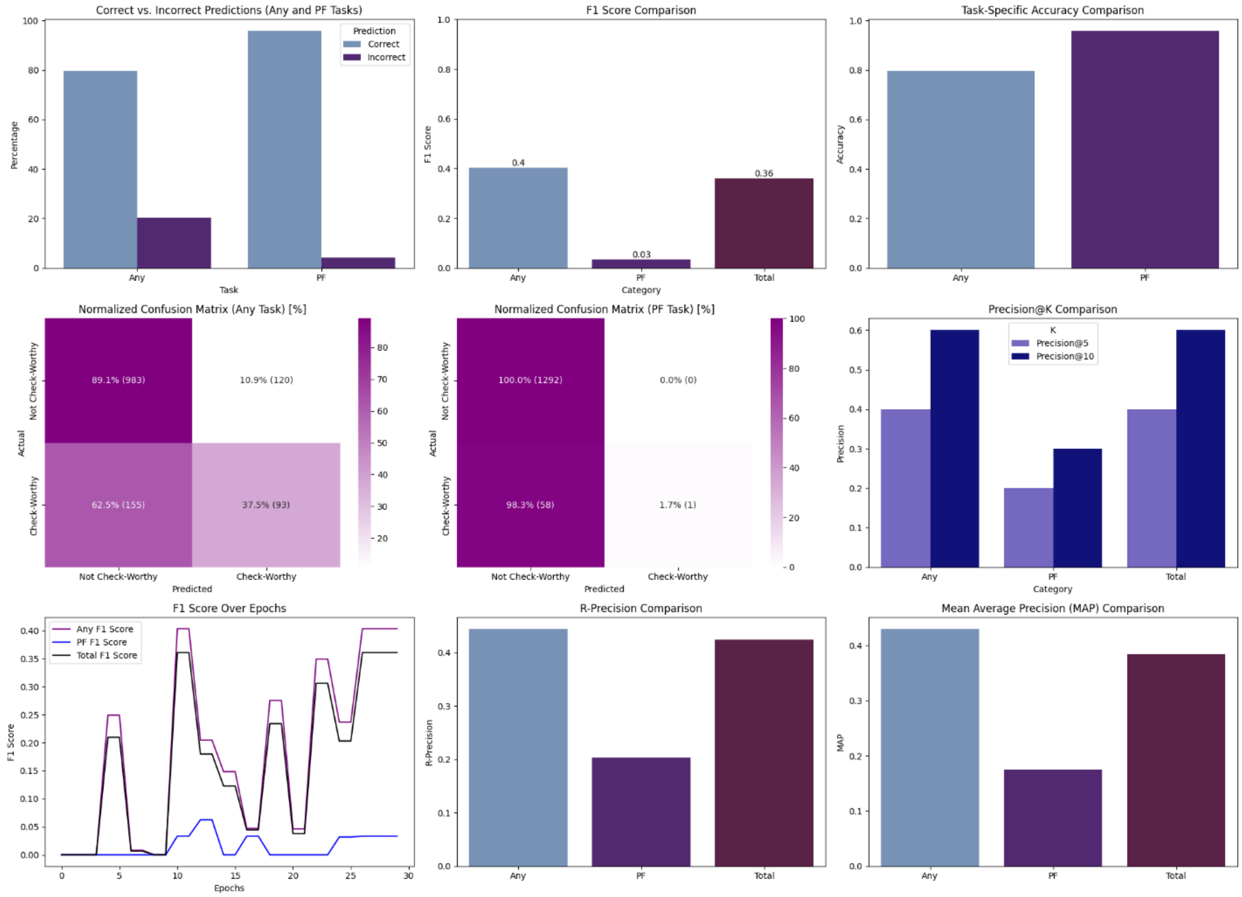
Precision @ K Comparison: The Precision @ K (for K=5 and K=10) for the "Any" task shows some improvement over the baseline, reaching around 0.4 and 0.7 respectively. The Precision @ K for the "Total" category is also slightly higher. This suggests a slightly better ability to retrieve relevant claims in the top predictions compared to the baseline.

F1 Score Over Epochs: The F1 score for the "Any" task shows some fluctuation and a gradual increase over epochs, reaching a higher final value than the baseline. The "Total" F1 score also shows a similar trend. The PF score remains low and shows poor increases. This indicates that the un-weighted approach, with average loss validation, allows for some learning, although the final F1 scores remain low.

R-Precision Comparison: The R-Precision for the "Any" category has improved to around 0.44, and the "Total" R-Precision is also slightly higher than the baseline. This suggests a marginal improvement in retrieving truly worth-checking claims within the top ranks.

Mean Average Precision (MAP) Comparison: The final MAP comparison shows a slight increase in the MAP score for the "Any" task compared to the baseline. The MAP score for the "PF" task remains very low, and the "Total" MAP score shows a modest improvement.

MultiTaskTransformer with Combined Score Validations Results



This section analyzes the performance of the Transformer model under the Task Un-Weighted treatment, where the losses for the "Any" and "PF" tasks were equally weighted during training, and the evaluation is based on a combined score utilizing both F1 score and average loss on the validation set.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart shows a similar pattern to previous treatments. For the "Any" task, a large majority of instances are predicted correctly, with a smaller portion incorrect. The "PF" task exhibits a very high rate of correct predictions.

F1 Score Comparison: The F1 score for the "Any" task has significantly improved to approximately 0.40 compared to the average loss validation setting. The F1 score for the "Total" task is around 0.36, also a substantial increase. And the F1 score for the “PF” task is 0.03. This indicates that evaluating based on a combined score has positively and considerably impacted the Transformer model's ability to balance precision and recall for identifying worth-checking claims.

Task-Specific Accuracy Comparison: The accuracy for the "Any" task is around 0.8, like the average loss validation setting. The accuracy for the "PF" task remains very high, approximately 0.98.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task shows that when the actual label was "Not Check-Worthy," the model correctly predicted it 89.1% of the time. When the actual label was "Check-Worthy," the model correctly identified it 37.5% of the time, with 62.5% being misclassified as "Not Check-Worthy”. This represents a substantial improvement in identifying "Check-Worthy" claims compared to the average loss validation setting.

Normalized Confusion Matrix (PF Task): The confusion matrix for the "PF" task shows a high correct prediction rate for "Not Check-Worthy" (100.0%) instances but a lower correct prediction rate for "Check-Worthy" (1.7%) instances. This suggests that while the model is good at identifying the majority class for the "PF" task, it still struggles with the minority class.

Precision @ K Comparison: The Precision @ K (for K=5 and K=10) for the "Any" task is estimated around 0.4 and 0.6 respectively. The Precision @ K for the "Total" task is also similar. This indicates a much better ability to retrieve relevant worth-checking claims within the top predictions than the baseline.

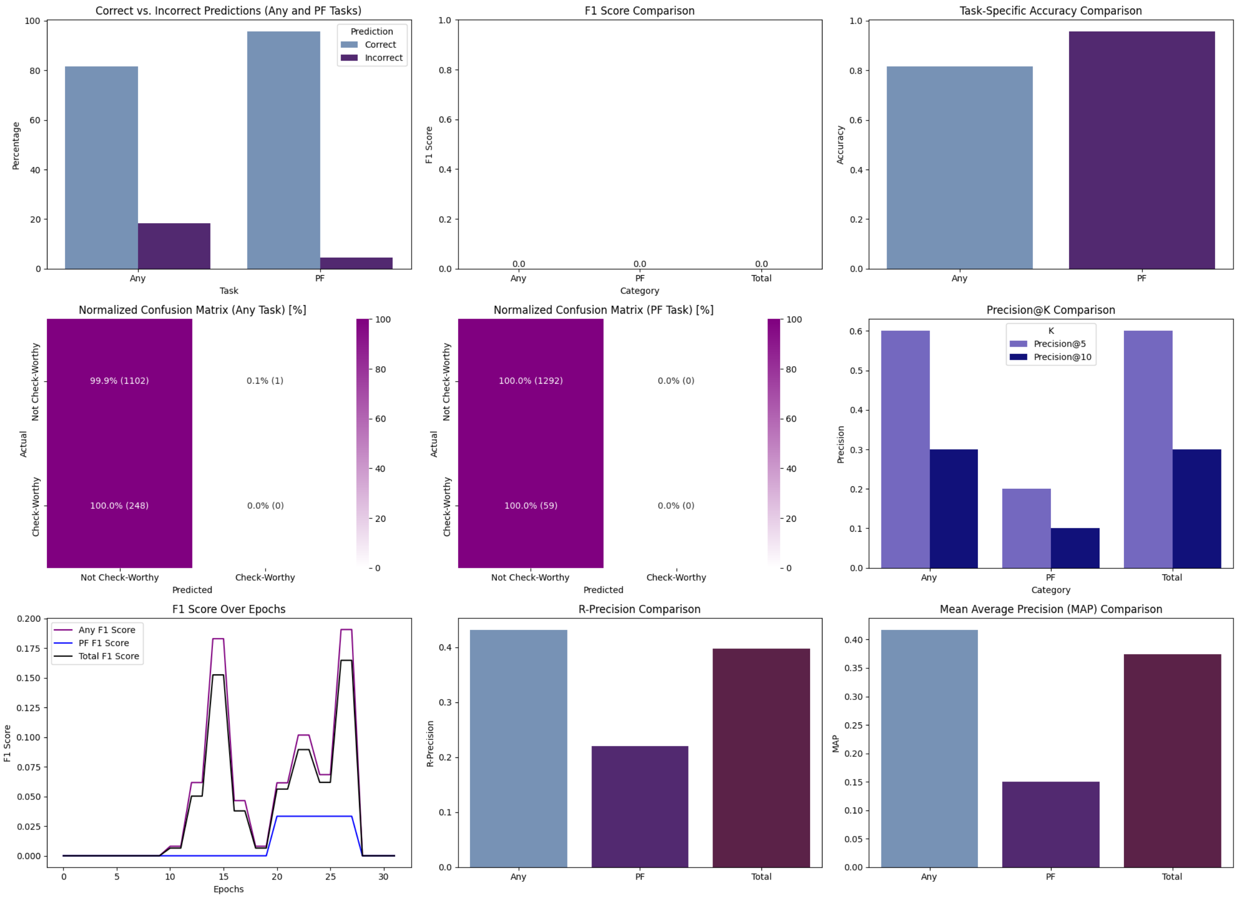
F1 Score Over Epochs: The F1 score for the "Any" task shows a more robust and higher learning curve over the epochs compared to the average loss validation, reaching a substantially higher final value. The "Total" F1 score also shows a significant improvement. This suggests that the combined score validation provides a more effective training signal for the Transformer model.

R-Precision Comparison: The R-Precision for the "Any" task is around 0.44, and the "Total" R-Precision is around 0.4, while PF remains low at 0.2. This confirms a similar ability to retrieve truly worth-checking claims within the top ranks.

Mean Average Precision (MAP) Comparison: The final MAP comparison shows a similar MAP score for the "Any" task compared to the average loss validation setting. The MAP score for the "Total" task is also similar. This indicates that the model is similar at ranking worth-checking claims higher in its predictions.

Task Weighted Treatment

MultiTaskRNN with Average Loss Validation Results



The results for the Task Weighted RNN treatment, evaluated using average validation loss, provide insight into the impact of prioritizing the "PF" task during training.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart indicates a strong bias in predictions. A large majority of instances are predicted correctly, especially for the "PF" task, while a smaller portion is incorrect.

F1 Score Comparison: The F1 score for the "Any" task is very low, around 0.0. The F1 score for the "Total" category is also very low, near 0.0. These low scores indicate poor performance in balancing precision and recall for both "Any" and "Total" predictions.

Task-Specific Accuracy Comparison: Accuracy for the "Any" task is relatively high, around 0.8. Accuracy for the "PF" task is very high, around 0.98. The high "PF" accuracy suggests the model is likely prioritizing the majority class.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task shows that when the actual label was "Not Check-Worthy," the model correctly predicted it 99.9% of the time. However, when the actual label was "Check-Worthy," the model incorrectly classified all instances (0.0% correct).

Normalized Confusion Matrix (PF Task): The confusion matrix for the "PF" task shows 100.0% correct predictions for both "Not Check-Worthy" and 0.0% for "Check-Worthy" instances. This accuracy is likely due to predicting the primary class correctly.

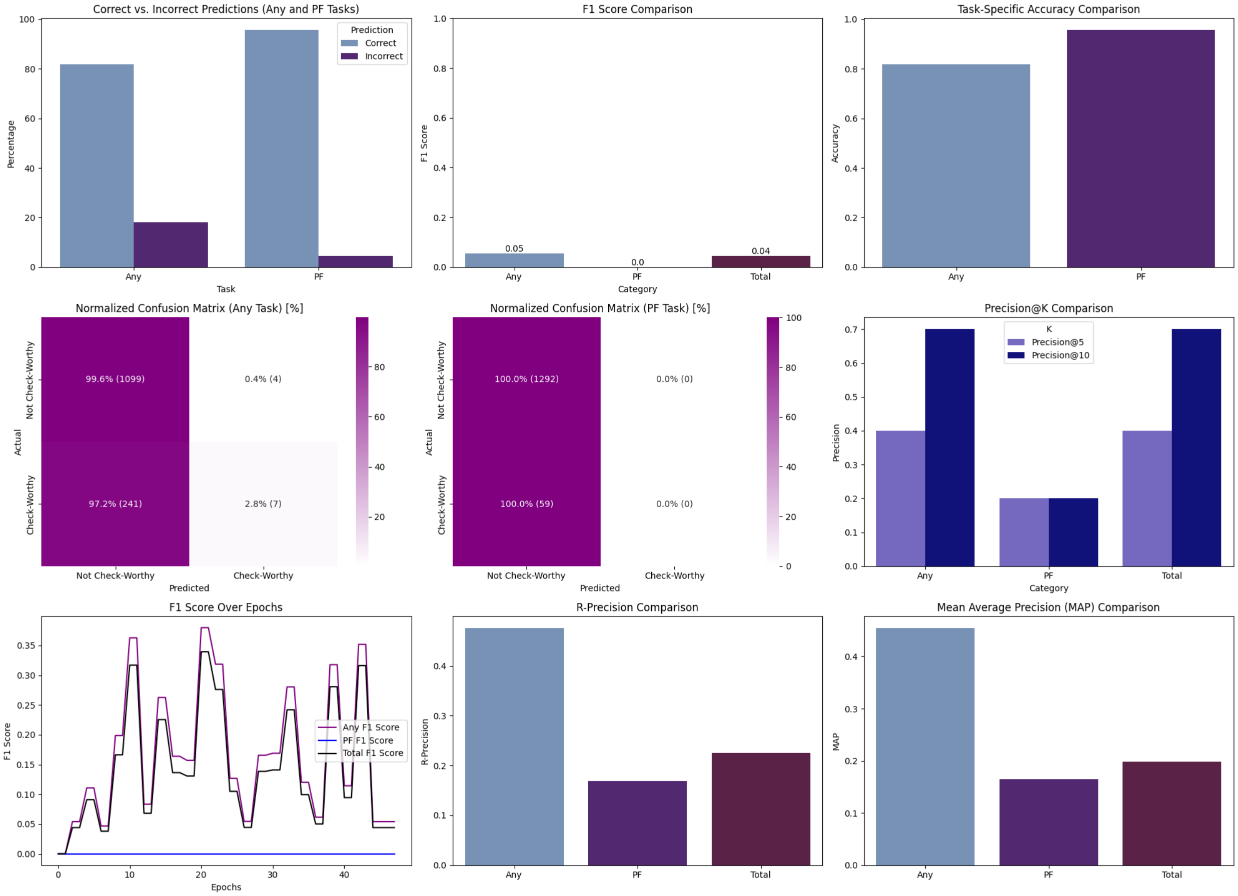
Precision @ K Comparison: Precision @ K (for K=5 and K=10) is 0.6 and 0.3 for the "Any" task. Precision @ K is similar low for the "Total" task. The “PF” task shows abysmal scoring at 0.2 and 0.1, respectively. This shows the model's inability to retrieve relevant worth-checking claims well.

F1 Score Over Epochs: The F1 score for the "PF" task remains at 0.0 throughout most epochs, with a slight increase at the end. The F1 score for the "Any" task starts low and shows a minor increase towards the end. The "Total" F1 score mirrors the "Any" task.

R-Precision Comparison: R-Precision for the "Any" task is 0.4. R-Precision is also very low for the "Total" task, at 0.4. And poor on the “PF” task at 0.21. This confirms the model's poor retrieval performance.

Mean Average Precision (MAP) Comparison: The final MAP comparison shows a MAP score of 0.4 for the "Any" task. The MAP score for the "PF" task remains very low. The "Total" MAP score is also low. This is similar to the baseline.

MultiTaskRNN with Combined Score Validations Results



The results for the Task Weighted RNN treatment, evaluated with combined score validations, provide insights into the impact of prioritizing the "PF" task during training when using a combined score evaluation.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart indicates a strong bias in predictions. A large majority of instances are predicted correctly, especially for the "PF" task, while a smaller portion is incorrect.

F1 Score Comparison: The F1 score for the "Any" task is very low, around 0.05. The F1 score for the "Total" task is also very low, around 0.04. The “PF” task shows a 0.0. These low scores indicate poor performance in balancing precision and recall, even with task weighting.

Task-Specific Accuracy Comparison: Accuracy for the "Any" task is relatively high, around 0.8. Accuracy for the "PF" task is very high, around 0.98. The high "PF" accuracy again suggests the model is likely prioritizing the majority class.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task shows that when the actual label was "Not Check-Worthy," the model correctly predicted it 99.6% of the time. However, when the actual label was "Check-Worthy," the model incorrectly classified a large majority of instances (97.2% misclassified).

Normalized Confusion Matrix (PF Task): The confusion matrix for the "PF" task shows 100.0% correct predictions for "Not Check-Worthy" and 100% incorrect predictions for "Check-Worthy" instances. This opposition is likely due to the primary class being identified.

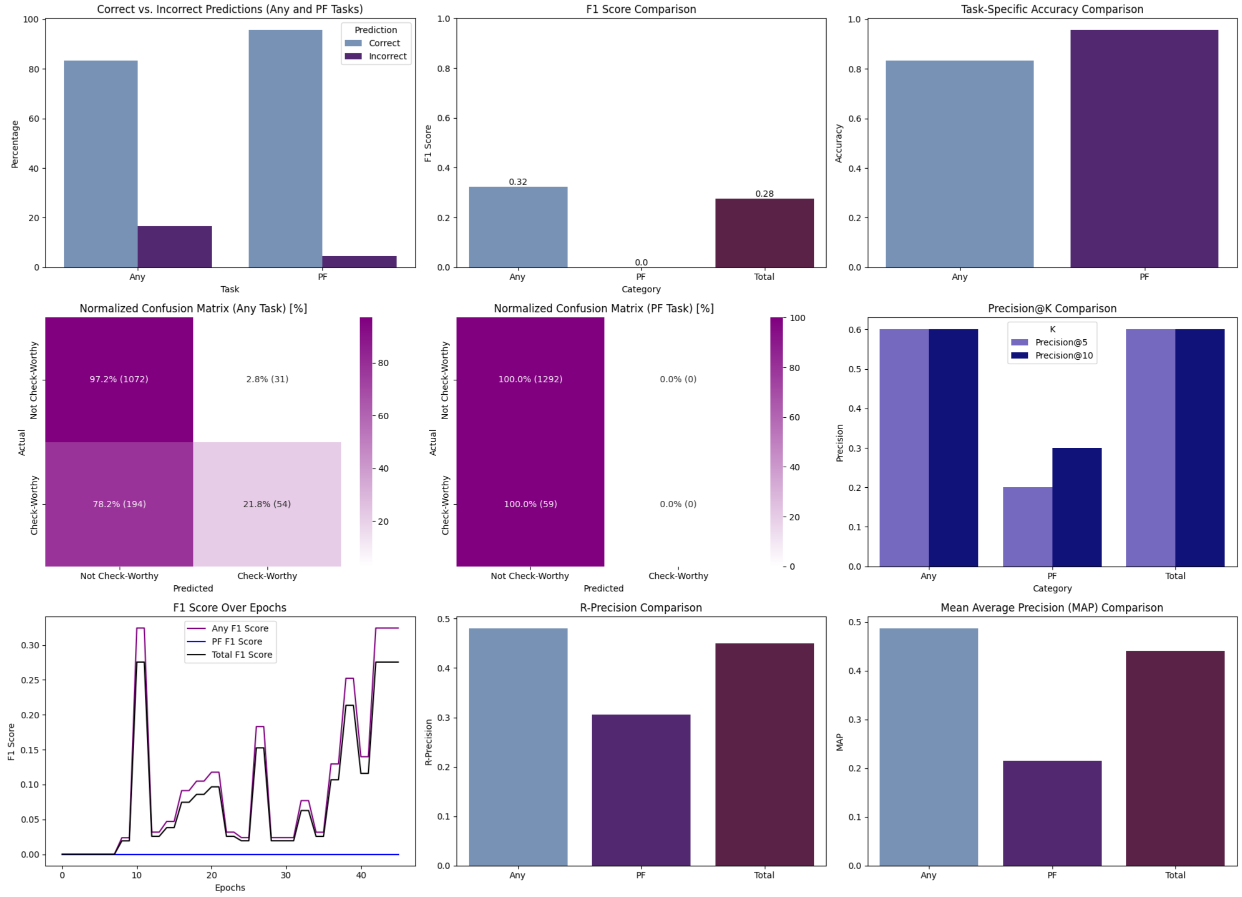
Precision @ K Comparison: Precision @ K (for K=5 and K=10) is very low for the "Any" task at 0.4 and 0.7, respectively. Precision @ K is also similar score for the "Total" task. The “PF” task shows the lowest scores at 0.2 for both. This shows the model's inability to retrieve relevant worth-checking claims.

F1 Score Over Epochs: The F1 score for the "Any" task remains consistently low with minimal improvement throughout the epochs. The F1 score for the "PF" task remains very low, never showing improvement. The "Total" F1 score shows minimal improvement over epochs.

R-Precision Comparison: R-Precision for the "Any" category is very low. R-Precision is also very low for the "Total" task. This confirms the model's poor retrieval performance.

Mean Average Precision (MAP) Comparison: The final MAP comparison shows a MAP score of 0.4 for the "Any" task. The MAP score for the "PF" task is very low at 0.18. The "Total" MAP score is also very low at 0.2.

MultiTaskTransformer with Average Loss Validation Results



The results for the Task Weighted Transformer treatment, evaluated with average loss, provide insights into the impact of prioritizing the "PF" task during training on the Transformer model.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart indicates a strong bias in predictions. A large majority of instances are predicted correctly, especially for the "PF" task, while a smaller portion is incorrect.

F1 Score Comparison: The F1 score for the "Any" task is approximately 0.32. The F1 score for the "Total" task is around 0.28. The “PF” task still shows a 0.0. These F1 scores show some improvement compared to the unweighted Transformer, but they still indicate a poor balance between precision and recall.

Task-Specific Accuracy Comparison: Accuracy for the "Any" task is relatively high, around 0.81. Accuracy for the "PF" task is very high, around 0.99. The high "PF" accuracy again suggests the model is likely prioritizing the majority class.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task shows that when the actual label was "Not Check-Worthy," the model correctly predicted it 97.2% of the time. However, when the actual label was "Check-Worthy," the model misclassified a large majority of instances (78.2% misclassified).

Normalized Confusion Matrix (PF Task): The confusion matrix for the "PF" task shows 100.0% correct predictions for "Not Check-Worthy" and 100% incorrect predictions for "Check-Worthy" instances. This is likely due to the primary class being classified every time.

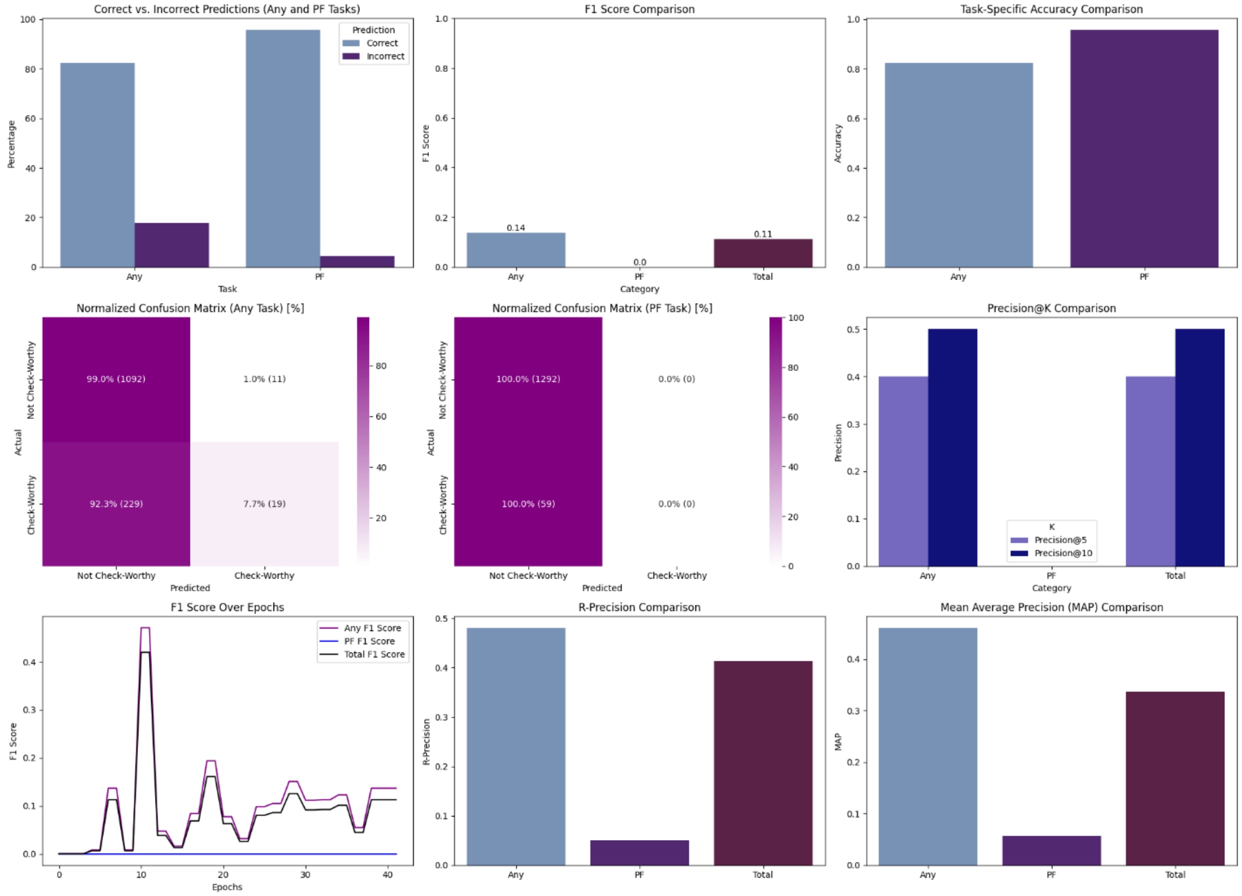
Precision @ K Comparison: Precision @ K (for K=5 and K=10) shows some improvement compared to the unweighted Transformer but is still not high.

F1 Score Over Epochs: The F1 score for the "Any" task shows some fluctuation and a gradual increase over epochs. The "Total" F1 score also shows a similar trend. The “PF” F1 score contains to show no improvement.

R-Precision Comparison: R-Precision is higher than the un-weighted Transformer, but still not high.

Mean Average Precision (MAP) Comparison: The final MAP comparison shows a modest increase in the MAP score for the "Any" task compared to the unweighted Transformer. The MAP score for the "PF" task remains low. The "Total" MAP score is also moderately improved.

MultiTaskTransformer with Combined Score Validations Results



The results for the Task Weighted Transformer treatment, evaluated with combined score validations, provide insights into the impact of prioritizing the "PF" task during training on the Transformer model when using a combined score evaluation.

Correct vs. Incorrect Predictions (Any and PF Tasks): The bar chart indicates a strong bias in predictions. A large majority of instances are predicted correctly, especially for the "PF" task, while a smaller portion is incorrect.

F1 Score Comparison: The F1 score for the "Any" task is approximately 0.14. The F1 score for the "Total" category is approximately 0.11. These F1 scores are low, suggesting poor performance in balancing precision and recall, even with task weighting and combined score evaluation.

Task-Specific Accuracy Comparison: Accuracy for the "Any" task is relatively high, around 0.81. Accuracy for the "PF" task is very high, around 0.98. The high "PF" accuracy again suggests the model is likely prioritizing the majority class.

Normalized Confusion Matrix (Any Task): The confusion matrix for the "Any" task shows that when the actual label was "Not Check-Worthy," the model correctly predicted it 99.0% of the time. However, when the actual label was "Check-Worthy," the model incorrectly classified a large majority of instances (92.3% misclassified).

Normalized Confusion Matrix (PF Task): The confusion matrix for the "PF" task shows 100.0% correct predictions for "Not Check-Worthy" and 100% incorrect predictions for "Check-Worthy" instances. This is likely due to assigning the primary class each time.

Precision @ K Comparison: Precision @ K (for K=5 and K=10) is low for the "Any" and "Total" tasks. But nonexistent for the “PF” task. This shows a poor ability to retrieve relevant worth-checking claims.

F1 Score Over Epochs: The F1 score for the "PF" task remains consistently very low throughout the epochs. The F1 score for the “Any” and "Total" F1 Score shows minimal improvement over epochs.

R-Precision Comparison: R-Precision is low for both the "Any" and "Total" tasks, and very low for “PF” task. This confirms the model's poor retrieval performance.

Mean Average Precision (MAP) Comparison: The final MAP comparison shows very low MAP scores for the "Any" and "PF" tasks. The "Total" task MAP score is also low.

Analysis and Discussion

Summary of Baseline RNN Performance

The Baseline RNN results, assessed via average validation loss, reveal significant limitations in effectively identifying worth-checking claims. While exhibiting high accuracy (around 80% for "Any" and near 100% for "PF"), this is largely attributed to a strong bias towards predicting the majority "Not Check-Worthy" class, as evidenced by the normalized confusion matrices. Specifically, for the "Any" task, the model never correctly identified "Check-Worthy" instances, and the "PF" task's seemingly perfect accuracy is also suspected due to the class imbalance. This bias is further underscored by the extremely low F1 scores (around 0.0 for "Any" and "Total"), indicating a failure to balance precision and recall. Moreover, the low Precision @ K (around 0.3 for "Any" and "Total", and 0.1 for "PF") suggests that even among the model's top predictions, the proportion of truly worth-checking claims is poor. Finally, the consistently low MAP scores (below 0.35 for "Any", near 0.0 for "PF" with minimal improvement, and low for "Total") and R-Precision (around 0.29 for "Any" and 0.27 for "Total") demonstrate the model's inability to effectively rank worth-checking claims higher, hindering its retrieval performance.

Summary of Baseline Transformer Performance

The Baseline Transformer, evaluated on average validation loss, demonstrates high accuracy (around 80% for "Any" and near 100% for "PF”), suggesting a tendency to correctly classify instances. However, the relatively low F1 scores (0.27 for "Any" and 0.23 for "Total") indicate a poor balance between precision and recall in identifying worth-checking claims. While the "Any" task's confusion matrix shows an improved 16.9% correct identification of "Check-Worthy" instances, there's still a large bias towards "Not Check-Worthy" (98.2% correct). The "PF" task's seemingly perfect confusion matrix is likely misleading due to class imbalance. Precision @ K scores are moderate for "Any" and "Total" (around 0.6-0.8) but poor for "PF" (0.3-0.4), indicating a limited proportion of truly worth-checking claims in the top predictions. Furthermore, the MAP scores remain low across tasks ("Any" below 0.45, "PF" around 0.2, and "Total" below 0.4), and R-Precision is also poor (0.44 for "Any", 0.4 for "Total", and 0.22 for "PF"), highlighting the model's difficulty in effectively ranking and retrieving truly worth-checking claims.

Summary of Task Un-Weighted RNN Performance (Average Loss Validation)

The MultiTaskRNN with unweighted tasks, evaluated by average validation loss, mirrors the Baseline RNN's tendency towards majority class prediction. While achieving high accuracy (around 80% for "Any" and near 100% for "PF"), this is driven by a strong bias, evidenced by the "Any" task's confusion matrix where "Check-Worthy" instances were never correctly identified. The F1 scores for "Any" (0.0) and "Total" (near 0.0) are the worst case of the baseline, indicating a detrimental impact of equal task weighting on balanced prediction. Precision @ K is high for "Any" at K=5 (1.0) but drops significantly at K=10 (0.6), and remains very low for "PF", suggesting poor retrieval of truly worth-checking claims. Similarly, F1 scores over epochs remain consistently low for "Any" and "Total". The R-Precision (0.43 for "Any", 0.41 for "Total", and 0.2 for "PF") and final MAP scores (0.4 for "Any" and "Total", 0.2 for "PF") further confirm the model's ineffective retrieval and ranking of worth-checking claims under this unweighted treatment.

Summary of Task Un-Weighted RNN Performance (Combined Score Validations)

Evaluating the unweighted MultiTaskRNN using a combined F1 score and average loss metric on the validation set yields noticeable improvements, particularly for the "Any" task. While maintaining high accuracy (around 80% for "Any" and 97% for "PF"), the F1 score for "Any" significantly increases to 0.26 (from 0.0), and "Total" reaches 0.22, suggesting a better balance of precision and recall, though further improvement is needed. The confusion matrix for "Any" shows a substantial gain in correctly identifying "Check-Worthy" instances (17.3%). Similarly, "PF" task prediction shows a slightly improved ability to identify both classes. Precision @ K for "Any" and "Total" dramatically improves (around 0.8 and 0.7 for K=5 and K=10), indicating a higher proportion of truly worth-checking claims in the top predictions, though "PF" still lags significantly. The F1 score over epochs for "Any" and "Total" shows a more dynamic and ultimately better learning trend. R-Precision also improves for "Any" (around 0.44) and "Total" (around 0.37), indicating better retrieval of relevant claims. However, MAP scores remain consistent with the average loss validation, suggesting the combined score primarily impacts the balance of precision and recall rather than the overall ranking ability.

Summary of Task Un-Weighted Transformer Performance (Average Loss Validation)

Evaluating the unweighted MultiTaskTransformer using average validation loss reveals a persistent bias towards the majority class. While accuracy remains high (around 80% for "Any" and 99% for "PF"), F1 scores for "Any" (0.15) and "Total" (0.12) show only marginal improvements over the baseline, indicating a continued poor balance between precision and recall. The "Any" task's confusion matrix still highlights a strong bias, with only 8.5% of "Check-Worthy" instances correctly identified. The "PF" task's confusion matrix suggests the model predicts only the majority class. Precision @ K shows some improvement for "Any" (0.4 at K=5, 0.7 at K=10) and "Total", suggesting a slightly better retrieval of relevant claims in top predictions. F1 scores over epochs show a gradual increase for "Any" and "Total", indicating some learning, but final scores remain low, and "PF" performs poorly. R-Precision for "Any" (0.44) and "Total" also sees a slight increase. Similarly, MAP scores show modest improvements for "Any" and "Total" but remain very low for "PF". Overall, while un-weighted training with average loss validation allows for some learning in the Transformer, the performance in effectively identifying and ranking worth-checking claims remains limited, with a strong bias towards the majority class.

Summary of Task Un-Weighted Transformer Performance (Combined Score Validations)

Evaluating the unweighted MultiTaskTransformer using a combined F1 score and average loss metric on the validation set leads to considerable improvements, particularly for the "Any" and "Total" tasks. While maintaining high accuracy for "Any" (around 0.8) and "PF" (around 0.98), the F1 score for "Any" significantly increases to approximately 0.40 (from 0.15 in the average loss setting), and "Total" rises to around 0.36 (from 0.12). However, the "PF" task's F1 score remains very low at 0.03. The "Any" task's confusion matrix shows a substantial improvement in correctly identifying "Check-Worthy" instances (37.5%). While the "PF" task achieves perfect accuracy for "Not Check-Worthy" (100.0%), it struggles significantly with "Check-Worthy" instances (only 1.7% correct). Precision @ K for "Any" is estimated at around 0.4 for K=5 and 0.6 for K=10, with similar values for "Total," indicating a much better retrieval of relevant claims in the top predictions compared to the average loss setting. The F1 scores over epochs for "Any" and "Total" show a more robust and higher learning curve. R-Precision for "Any" is around 0.44 and for "Total" around 0.4, while "PF" remains low at 0.2, suggesting a similar ability to retrieve truly worth-checking claims within the top ranks. The final MAP scores for "Any" and "Total" are similar to the average loss validation, indicating a consistent ranking ability.

Summary of Task Weighted RNN Performance (Average Loss Validation)

The Task Weighted RNN, prioritizing the "PF" task and evaluated by average validation loss, exhibits a strong predictive bias, particularly towards the majority class in the "PF" task. While achieving relatively high accuracy for "Any" (around 0.8) and "PF" (around 0.98), the F1 scores for both "Any" and "Total" remain very low (around 0.0), indicating a poor balance between precision and recall. The confusion matrix for "Any" reveals that "Check-Worthy" instances were never correctly identified, and the "PF" task shows a similar issue, likely predicting only the majority class. Precision @ K is moderate for "Any" (0.6 at K=5, 0.3 at K=10) but abysmal for "PF" (0.2 at K=5, 0.1 at K=10), suggesting a poor ability to retrieve relevant worth-checking claims. F1 scores over epochs remain near zero for "PF" and show only a slight late increase for "Any" and "Total". R-Precision is low for "Any" (0.4) and "Total" (0.4), and very poor for "PF" (0.21), confirming poor retrieval performance. The final MAP scores are also low across tasks, similar to the baseline, indicating no improvement in ranking worth-checking claims. Prioritizing the "Any" task in this manner did not improve the model's ability to effectively identify or rank worth-checking claims.

Summary of Task Weighted RNN Performance (Combined Score Validations)

Evaluating the Task Weighted RNN with combined score validation, prioritizing the "PF" task, still reveals a strong predictive bias towards the majority class, particularly evident in the "PF" task. Despite high accuracy for "Any" (around 0.8) and "PF" (around 0.98), the F1 scores for "Any" (0.05), "Total" (0.04), and "PF" (0.0) remain very low, indicating a persistent failure to balance precision and recall. The "Any" task's confusion matrix shows that a large majority (97.2%) of "Check-Worthy" instances were misclassified. The "PF" task exhibits a complete failure to identify "Check-Worthy" instances. Precision @ K is low for "Any" (0.4 at K=5, 0.7 at K=10) and similarly low for "Total," with "PF" showing the lowest scores (0.2 for both). F1 scores over epochs remain consistently low for all tasks, showing minimal improvement. R-Precision is also very low for "Any" and "Total," and poor for "PF," confirming poor retrieval performance. The final MAP scores are low for "Any" (0.4), "PF" (0.18), and "Total" (0.2). Thus, even with combined score validation and task weighting, the RNN model struggles to effectively identify and rank worth-checking claims, maintaining a strong bias and poor F1 scores.

Summary of Task Weighted Transformer Performance (Average Loss Validation)

Prioritizing the "PF" task in the Transformer model with average loss evaluation leads to some improvement compared to the un-weighted approach. While maintaining high accuracy for "Any" (around 0.81) and "PF" (around 0.99), the F1 scores for "Any" (0.32) and "Total" (0.28) show a modest increase but still indicate a poor balance between precision and recall. The "Any" task's confusion matrix reveals that a large proportion (78.2%) of "Check-Worthy" instances are still misclassified. The "PF" task continues to show a complete failure in identifying "Check-Worthy" instances. Precision @ K shows some improvement but remains not high. F1 scores over epochs for "Any" and "Total" show a gradual increase, but "PF" remains at zero. R-Precision is also higher than the unweighted setting but still not high. Similarly, MAP scores show a modest increase for "Any" and "Total" but remain low for "PF". Thus, task weighting with average loss provides some benefit to the Transformer's performance, but significant challenges in balancing precision and recall and correctly identifying the minority class persist.

Summary of Task Weighted Transformer Performance (Combined Score Validations)

Evaluating the Task Weighted Transformer with combined score validation, prioritizing the "PF" task, still results in low performance. While accuracy for "Any" (around 0.81) and "PF" (around 0.98) remains high, the F1 scores for "Any" (0.14) and "Total" (0.11) are low, indicating a persistent issue with balancing precision and recall. The "Any" task's confusion matrix reveals a high misclassification rate (92.3%) for "Check-Worthy" instances. The "PF" task continues to completely fail in identifying "Check-Worthy" instances. Precision @ K is low for "Any" and "Total" and nonexistent for "PF," demonstrating a poor ability to retrieve relevant claims. F1 scores over epochs show minimal improvement for all tasks. R-Precision is low for "Any" and "Total" and very low for "PF," confirming poor retrieval. The final MAP scores are also very low across all tasks. Thus, even with task weighting and combined score validation, the Transformer model struggles to effectively identify and rank worth-checking claims, maintaining a strong bias and poor F1 scores.

Conclusion

The Task Un-Weighted MultiTaskTransformer with Combined Score Validation showed the most promising results.

Let’s explore why based on the results:

Improved F1 Scores: This model and evaluation setting yielded the highest F1 scores for the "Any" (approximately 0.40) and "Total" (around 0.36) tasks compared to all other models and settings. This indicates a better balance between precision and recall in identifying worth-checking claims.

Better Identification of "Check-Worthy" Claims: The confusion matrix for the "Any" task in this setting showed a significantly higher correct identification rate for "Check-Worthy" instances (37.5%) compared to the other Transformer and RNN configurations.

Enhanced Precision @ K: The Precision @ K values for the "Any" and "Total" categories were notably higher in this setting (around 0.4-0.6 for "Any" and similar for "Total"), suggesting that a larger proportion of the top predicted claims were actually worth checking.

More Robust Learning: The F1 score over epochs showed a more dynamic and higher learning curve, indicating that the combined score validation provided a more effective training signal for the Transformer.

Comparable or Improved Retrieval: The R-Precision for "Any" was comparable to the best results of other models, suggesting a decent ability to retrieve truly worth-checking claims within the top ranks.

However, it's crucial to note the following limitations and considerations:

Persistent Issues with the "PF" Task: All models and settings struggled significantly with the "PF" task, consistently showing very low F1 scores, Precision @ K, and often failing to identify "Check-Worthy" instances. This suggests that the "PF" task presents a unique challenge that requires further investigation and potentially different modeling approaches or data handling.

Accuracy vs. F1 Score: While accuracy was generally high across many models, particularly for the "PF" task, this was often misleading due to class imbalance, as highlighted by the low F1 scores. Therefore, F1 score is a more reliable metric for evaluating the model's ability to identify worth-checking claims effectively.

MAP Scores: The Mean Average Precision (MAP) scores, which evaluate the ranking of worth-checking claims, did not show substantial improvement in the MultiTaskTransformer with combined score validation compared to other settings. This suggests that while the model improved in balancing precision and recall, its ability to rank positive instances higher did not see a similar level of gain.

Task Weighting Impact: Task weighting with average loss validation did show some improvement for the Transformer compared to the unweighted setting, but the combined score validation proved more effective. For the RNN, task weighting generally did not lead to better performance.

While the **MultiTaskTransformer with Combined Score Validation** appears to be the most promising model and evaluation strategy based on the results, particularly for the "Any" and "Total" tasks, there are still significant challenges, especially with the "PF" task and the overall ranking performance. Further research and optimization are needed to address these limitations and potentially improve the performance across all tasks.

In light of the substantial influence of the evaluation metric and task weighting identified in this study, promising avenues for future research include the exploration of advanced evaluation methodologies that extend beyond basic average loss calculations. Investigating further dynamic or adaptive task weighting mechanisms, which adjust the importance of each task during training based on performance, could also prove beneficial.

Furthermore, the application of specialized loss functions designed to address class imbalance, such as focal loss, warrants investigation, particularly for enhancing performance on the "PF" task. Finally, exploring variations within the RNN and Transformer architectures, potentially including hybrid models, coupled with comprehensive hyperparameter optimization, holds the potential to yield further improvements in the accurate identification and effective ranking of worth-checking claims across both evaluation targets.

Works Cited

Ash, E., Gauthier, G., & Widmer, P. (2024). Relatio: Text Semantics Capture Political and Economic Narratives. *Political Analysis*, 115-132. doi:https://doi.org/10.1017/pan.2023.8

Chen, W., Pacheco, D., Yang, K.-C., & Menczer, F. (2021). Neutral Bots Probe Political Bias on Social Media. *Nature Communications, 12*, 5580. doi:https://doi.org/10.1038/s41467-021-25738-6

Gencheva, P., Koychev, I., Màrquez, L., Barrón-Cedeño, A., & Nakov, P. (2017). A Context-Aware Approach for Detecting Check-Worthy Claims in Political Debates. *Proceedings of Recent Advances in Natural Language Processing* (pp. 267–276). Varna: INCOMA Ltd. doi:https://doi.org/10.26615/978-954-452-049-6_037