POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Conspiracy Theories vs Critical Thinking Using an Ensemble of Transformers and Large Language Models

Tesi di Laurea Magistrale in
Computer Science and Engineering - Ingegneria Informatica

Author: **Angelo Maximilian Tulbure**

Student ID: 217764
Advisor: Prof. Mark James Carman
Co-advisors: Prof. Paolo Rosso, Mariona Coll Ardanuy (UPV)
Academic Year: 2023-24

Abstract

This thesis contributes to the PAN at CLEF 2024 Oppositional Thinking Analysis shared task, focusing on automatically differentiating between conspiratorial and critical thinking narratives. The task includes binary classification to identify texts as conspiratorial or critical and span-level detection to recognize narrative elements like Agents, Facilitators, Victims, Campaigners, Objectives, and Negative Effects. Two multilingual datasets, in English and Spanish, with 5,000 annotated Telegram comments each, presented challenges such as class imbalances and complexities in detecting partially overlapping spans.

For the binary classification task, a Soft Voting ensemble of fine-tuned Transformer models, enhanced with data augmentation techniques like back-translation, achieved F1-macro scores of 0.8917 for English and 0.8293 for Spanish.

The span-detection task emphasized precise tokenization and synonym replacement to maintain token alignment, crucial for detecting narrative spans. Great emphasis was placed on data preprocessing that further improved model robustness, achieving a span-F1 score of 0.6279 for English and 0.6129 for Spanish, which secured the best performance in the shared task for both languages and set a new state-of-the-art.

Additionally, experiments using Large Language Models (LLMs) significantly improved these results. By first employing zero-shot and few-shot learning techniques, and then fine-tuned LLMs enhanced cross-lingual understanding and narrative detection. These improvements underscore the effectiveness of LLMs in handling complex multilingual datasets, setting new performance benchmarks for both languages.

This thesis contributes to the field of narrative analysis, offering insights that may inform future research on improving narrative detection across other domains and highlighting the role of automated systems in combating misinformation in digital spaces.

Keywords: oppositional thinking analysis, conspiracy theories, critical thinking, binary classification, span-level detection, transformers, LLMs, ensembling models

Abstract in lingua italiana

La presente tesi contribuisce alla Oppositional Thinking Analysis shared task del PAN al CLEF 2024, concentrandosi sulla distinzione automatica tra narrazioni cospirazioniste e pensiero critico. Il task è suddiviso in una classificazione binaria per identificare i testi come cospirazionisti o critici e una rilevazione a livello di span per riconoscere elementi narrativi come Agenti, Facilitatori, Vittime, Attivisti, Obiettivi ed Effetti Negativi. Sono stati forniti due dataset multilingue, in inglese e spagnolo, ciascuno composto da 5.000 commenti annotati, che hanno presentato sfide come lo sbilanciamento delle classi e le complessità nella rilevazione di span parzialmente sovrapposti.

Per il task di classificazione binaria, un Soft Voting ensemble di fine-tuned Transformer models, potenziati con tecniche di aumento dei dati come la back-translation, ha ottenuto punteggi F1-macro pari a 0,8917 per l'inglese e 0,8293 per lo spagnolo.

Il task di rilevazione degli span ha enfatizzato la tokenizzazione precisa e la sostituzione tramite sinonimi per mantenere l'allineamento dei token, fondamentale per rilevare gli span narrativi. È stata posta grande enfasi sulla preelaborazione dei dati che ha ulteriormente migliorato la robustezza del modello, raggiungendo un punteggio span-F1 di 0,6279 per l'inglese e 0,6129 per lo spagnolo, garantendo i migliori risultati nel task per entrambe le lingue e stabilendo un nuovo riferimento per ricerche future. Inoltre, esperimenti con modelli di linguaggio di grandi dimensioni (LLM) hanno migliorato significativamente questi risultati. Utilizzando inizialmente tecniche di apprendimento zero-shot e few-shot, e successivamente facendo il fine-tuning dei LLM, è stata migliorata la comprensione interlinguistica e la rilevazione degli elementi narrativi. Questi miglioramenti evidenziano l'efficacia dei LLM nella gestione di dataset multilingue complessi, fissando nuovi standard di prestazione per entrambe le lingue. Questa tesi contribuisce al campo dell'analisi narrativa e sottolinea l'importanza dei sistemi automatizzati nella lotta alla disinformazione negli spazi digitali.

Parole chiave: analisi del pensiero oppositivo, teorie del complotto, pensiero critico, classificazione binaria, rilevamento a livello di span, transformers, LLMs, ensembling models

Contents

1 | Introduction

Conspiracy theories offer elaborate explanations for significant events, attributing them to hidden schemes orchestrated by secretive and powerful groups. On the other hand, critical thinking involves the careful evaluation of evidence, questioning assumptions, and forming reasoned judgments based on logical and evidence-based analysis. While both forms of thought may involve skepticism and questioning, they diverge in their goals and approaches. Critical thinking serves as a foundation for informed decision-making, challenging established norms through a structured examination of facts and reasoning. Conspiracy theories, instead, often rely on emotional appeal, selective use of evidence, and unverified claims, leading to distorted interpretations of reality. As a result, distinguishing between these two forms of thought is crucial for maintaining a healthy public discourse.

In today's digital age, the rise of social media platforms has created an environment in which both conspiracy theories and critical perspectives can spread rapidly. Conspiracy theories, in particular, have flourished on social platforms, where the lack of fact-checking mechanisms and the speed of information sharing allow misinformation and disinformation to gain traction. While misinformation encompasses unintentional spreading of incorrect information, disinformation refers to false information deliberately spread with the intent to deceive. The consequences of this phenomenon can be severe, with widespread public confusion and erosion of trust in institutions. At the same time, social media platforms also serve as spaces for critical thinkers to question prevailing narratives and engage in constructive debate, which is essential for promoting transparency and accountability in society.

One of the main challenges in detecting conspiracy theories lies in their linguistic complexity and the rhetorical strategies they share with critical thinking. Both conspiracy theorists and critical thinkers question authority, present alternative viewpoints, and challenge mainstream narratives. However, while conspiracy theories often distort facts and rely on sensationalism, critical thinking is driven by logic and the pursuit of truth. This overlap in rhetorical style makes it difficult for automated systems to distinguish between the two, which can lead to the misclassification of critical thinking as conspiratorial or

the failure to detect harmful misinformation.

A crucial aspect of narrative detection, beyond the binary classification of conspiracy and critical thinking, is the identification of key elements that structure these narratives. These elements include Agents, who are portrayed as the primary actors responsible for certain actions or outcomes within a narrative. In conspiracy theories, agents are often characterized as powerful entities or organizations working in secret to manipulate events. Facilitators are individuals or groups that assist the agents in achieving their objectives, acting as intermediaries or supporters. Victims are those who suffer as a result of the agents' actions, often depicted as the public or marginalized groups. Another essential element, Campaigners, are those who oppose the actions or narratives of the agents, frequently positioned as voices of resistance or opposition to the mainstream narrative. Objectives, in this context, represent the goals or intentions of the Agents, which in conspiracy narratives may involve the pursuit of political control, financial gain, or societal manipulation. Finally, negative effects refer to the adverse consequences experienced by the victims, which can range from public health crises to economic instability, depending on the narrative being propagated.

Recognizing these narrative elements is essential for understanding how both conspiracy theories and critical thinking are constructed and propagated. In many cases, the presence of these elements can be subtle, and they often overlap within a single narrative. For instance, a particular text might frame an individual as both a campaigner against the establishment and a victim of the actions taken by agents. This complexity poses additional challenges for automated systems, as distinguishing these elements within multilingual and cross-cultural contexts further complicates the detection process.

The shared task "Oppositional thinking analysis: Conspiracy theories vs critical thinking narratives" [48], part of PAN at CLEF 2024 [9], addresses the problem of detecting these narratives in text. The task involves both binary classification, where texts are classified as either conspiratorial or critical, and span-level detection, where the narrative elements mentioned earlier must be identified and categorized. The dataset provided for this task consists of 5,000 annotated Telegram comments in English and other 5,000 in Spanish, which presents the additional challenge of handling multilingual data and the diverse narrative structures that come with it.

Our approach to these tasks involves fine-tuning transformer models, which have demonstrated state-of-the-art performance in natural language processing tasks [97]. For the binary classification task, we employ a Soft Voting Ensembling method, which combines predictions from multiple transformer models to improve accuracy and robustness. The

class imbalance present in the dataset, with a greater number of critical thinking texts compared to conspiracy texts, is addressed through data augmentation techniques, such as back-translation, which help to diversify the training data and reduce bias.

For the span-level detection task, we treat the problem as a token classification task. This requires precise handling of tokenization and alignment to ensure that narrative spans are detected accurately. Given the presence of overlapping spans and the complexity of some texts, this task presents significant challenges.

Due to our work, our team earned the distinction of being the best-performing team in the international Shared Task on Oppositional Thinking Analysis: Conspiracy Theories vs. Critical Thinking Narratives, as part of the PAN 2024 Lab of the CLEF Initiative[1] [97].

This research not only contributes to the technical advancements in narrative detection but also holds broader societal significance. The ability to accurately distinguish between conspiracy theories and critical thinking is essential for fostering constructive debate and preventing the spread of harmful misinformation. By identifying the key narrative elements that underpin these forms of thought, we can better understand how oppositional narratives are constructed and how they impact public discourse. This work ultimately aims to support efforts in promoting a more informed, critical, and healthy dialogue in digital spaces.

1.1. Thesis Objectives

The purpose of this thesis is to address the increasingly urgent need to differentiate between conspiracy theories and critical thinking in textual data, particularly in the context of online discussions where these narratives often overlap. This research contributes to the ongoing development of models and techniques that not only classify texts but also analyze the finer elements of oppositional narratives. Specifically, the thesis aims to:

- Develop and fine-tune Transformer models to detect conspiracy theories and critical thinking narratives.
- Address the challenges posed by multilingual datasets, particularly in English and Spanish, focusing on improving performance across different languages and narrative structures.
- Implement span-level detection techniques that go beyond simple classification by

[1] https://pan.webis.de/clef24/pan24-web/oppositional-thinking-analysis.html#award

identifying and categorizing key narrative elements.

- Explore the role of Large Language Models in enhancing narrative detection.
- Incorporate techniques such as data augmentation to address class imbalance and improve model robustness.
- Provide insights into the broader implications of detecting oppositional narratives, highlighting the importance of these techniques in mitigating the spread of disinformation and promoting healthier public discourse in online environments.

This thesis provides a comprehensive framework for approaching the problem of oppositional thinking analysis and advances the state of the art in both binary classification and span-level detection of narratives. Concretely, we aim at answering the three following Research Questions (RQs):

RQ1. *How effectively can transformer models and Large Language Models distinguish between conspiracy theories and critical thinking narratives, and which factors impact their performance?*

Subquestions:

- *What are the main challenges in differentiating linguistic and rhetorical features between conspiratorial and critical thinking narratives?*
- *How do the performances of different models vary across languages?*

RQ2. *How feasible is it to accurately identify text spans that correspond to key elements within oppositional narratives?*

Subquestions:

- *What are the primary difficulties in detecting partially overlapping narrative elements in multilingual datasets?*
- *How does the context in which narrative elements appear influence their identification accuracy across different models?*

RQ3. *To what extent do data augmentation techniques enhance model performance across the two tasks?*

Subquestions:

- *Which data augmentation techniques contribute most significantly to improving model performance in binary classification and span-level detection?*

- *How does data augmentation impact the model's ability to generalize across new or unseen narrative structures?*

1.2. Structure of the Thesis

The rest of the thesis is organised as follows:

Chapter 2, *State of the Art*, lays the foundational knowledge necessary for understanding the current methodologies and technological advancements in misinformation and disinformation detection. The evolution from traditional machine learning to modern deep learning, sets the stage for subsequent chapters.

Transitioning from these foundational concepts, Chapter 3, *Evolution of Disinformation Detection Tasks*, traces the shift in focusing from profiling individuals who spread disinformation to a more nuanced analysis of the narratives themselves. This chapter enhances the discussion initiated in Chapter 2 by contextualizing the challenges and significance of the PAN 2024 challenge, bridging theoretical frameworks with practical research applications.

Following this contextual groundwork, Chapter 4, *Dataset Description*, introduces the datasets used in this study, focusing on their structure and the inherent challenges they pose, such as class imbalances and the complexities of narrative span detection. This detailed dataset description naturally leads into the practical applications and experimental setups discussed in the subsequent chapter.

Chapter 5, *Evaluation Metrics*, outlines the metrics employed to assess model performance, providing a comprehensive foundation for understanding the empirical results discussed in later chapters. This section ensures that readers grasp the criteria used to evaluate the effectiveness of the models.

In Chapter 6, *Experimental Framework*, the thesis transitions into a detailed discussion of the experimental setups used in the systems described in the following chapters.

Chapter 7, *Binary Classification of Conspirational vs. Critical Thinking*, delves into the binary classification methodologies, discussing in-depth the processes of data processing, data augmentation, model training, and extraction of patterns. The integration and performance evaluation of various classifiers, especially ensemble methods, pave the way for a more granular analysis discussed in the following chapter.

Chapter 8, *Span-level Detection of Narrative Elements*, builds directly on the complexities introduced in Chapter 7 by focusing on the token classification task. It provides an

exhaustive analysis of the technical challenges encountered and the innovative solutions implemented to ensure precise token alignment and the effective handling of partially overlapping narrative elements.

Chapter 9, *Results on the Test Dataset*, presents the results of the models obtained on the test dataset. This chapter offers a critical evaluation of the effectiveness and limitations of the models across the two tasks.

Chapter 10, *Error Analysis*, reflects on the findings from Chapter 9 by conducting a detailed error analysis. This examination provides insights into the potential improvements and directs the focus toward future research paths discussed in the final chapter.

Chapter11, *Experiments with LLMs and Multilingual Transformer Models*, explores the experiments conducted with Large Language Models (LLMs) and multilingual transformer models, assessing their performance across the two different languages and tasks. This chapter also includes a discussion on fine-tuning LLMs and adapting multilingual transformer models to tackle the challenges posed by narrative detection in both English and Spanish datasets.

Chapter 12, *Conclusion and Future Work*, encapsulates the research findings, reflecting on the broader implications for narrative analysis and the field of disinformation detection. This chapter sets the stage for future work by outlining promising research trajectories based on the compiled research.

Appendices offer additional methodological details and experimental results to bolster the transparency and reproducibility of the research.

2 | State of the Art

This chapter introduces the state of the art in the field of disinformation and misinformation detection, outlining both the preliminary techniques and the most effective current methods, particularly in the areas of oppositional thinking analysis, focusing on binary classification and span-level detection.

2.1. Disinformation vs Misinformation

In the digital era, the rapid spread of information has proven to be both a benefit and a challenge. While it enhances communication and broadens access to knowledge, it also accelerates the proliferation of false information. Among the most significant challenges are *Disinformation* and *Misinformation*, two related yet distinct concepts differentiated by their motivations and consequences. This section explores their definitions, impacts, and how they shape contemporary information ecosystems [54].

Disinformation is the deliberate creation and distribution of false information with the intent to deceive and manipulate audiences. This orchestrated effort typically exploits vulnerabilities in belief systems by blending fact with fiction, making the falsehoods appear more credible [32].

These campaigns often utilize sophisticated methods, including bots and trolls to amplify messages, fake news websites to create a facade of legitimacy, and social media algorithms to maximize dissemination. Targeted disinformation frequently preys on fears, biases, and uncertainties, making it an especially effective tool for dividing societies and eroding trust in institutions [90].

One prominent example is the use of state-funded propaganda to manipulate public opinion, particularly during elections. Disinformation can skew perceptions of political candidates by spreading false narratives or damaging personal information, leading to manipulated voting outcomes. It also can foment social unrest by promoting divisive content aimed at weakening social cohesion and institutional trust [3, 10].

A well-known case of disinformation is Russia's interference in the 2016 U.S. presidential

election, where the Internet Research Agency, a state-sponsored entity, systematically used fake accounts and misleading articles to manipulate public opinion [67].

Disinformation's impact extends beyond politics, infiltrating economic systems as well. False claims about companies or markets can cause volatility, affecting stock prices and financial stability. Moreover, the psychological phenomenon known as the "illusory truth effect" plays a significant role, where repeated exposure to false information can lead individuals to accept it as truth, further entrenching false beliefs and making corrective efforts difficult [14].

Unlike disinformation, misinformation is not intended to deceive, but its consequences can be just as damaging, as it can perpetuate false beliefs, undermine trust in reliable sources, and contribute to the spread of unfounded claims or panic, much like disinformation does. [39, 55].

Social media platforms have made it easy for misinformation to spread rapidly through sharing and resharing, often without proper fact-checking. This is particularly concerning during crises, where uncertainty prevails. For example the rapid spread of misinformation on social media about virus transmission and preventive measures during the COVID-19 pandemic severely undermined public health efforts [14, 72].

Cognitive biases, such as confirmation bias, further accelerate the spread of misinformation. Individuals are more likely to accept information that aligns with their pre-existing beliefs, dismissing information that contradicts them. In like-minded communities, this reinforces the cycle of misinformation, allowing it to spread unchecked and creating echo chambers where falsehoods thrive [22, 54].

The consequences of misinformation can be severe. In public health, for instance, misinformation about vaccines can lead to widespread vaccine hesitancy, which, in turn, undermines efforts to control preventable diseases. The spread of false information during health crises has shown how quickly misinformation can outpace authoritative responses, contributing to public confusion and the adoption of dangerous behaviors [16].

Addressing both disinformation and misinformation requires a multifaceted approach. For disinformation, regulatory measures are essential to hold platforms accountable, along with tools to detect and block disinformation campaigns before they spread. Increasing digital literacy among the public is equally critical, as it helps people recognize and resist false information.

Combatting misinformation, on the other hand, involves improving the transparency and accessibility of accurate information, especially during crises. Fact-checking services and

tools can make a meaningful difference in mitigating the spread, but they must be accompanied by efforts to reduce cognitive biases and improve critical thinking skills across society [64, 74].

2.2. Key Elements in Disinformation and Misinformation

In the context of disinformation and misinformation, understanding the various elements involved in false information campaigns is crucial. These elements shape the structure and impact of such campaigns, influencing how false information is created, propagated, and received. Key elements include:

Agents: Individuals or groups who actively create and disseminate false information. In the case of disinformation, agents often include state actors, political operatives, or organized groups with a clear agenda to deceive and manipulate. These agents use sophisticated techniques to ensure their narratives reach and influence their target audience. For example, during the 2016 US presidential election, Russian operatives acting as agents created and distributed false information to sway public opinion. These agents used fake social media profiles and automated bots to amplify their messages, making them appear more credible and widely supported than they actually were [10].

Facilitators: Those who assist agents in spreading false information. They may not be the originators but play a significant role in its propagation. Facilitators can include media organizations, social media platforms, or individuals who share and amplify false narratives without necessarily understanding their falsehood or intent. For instance, during various misinformation campaigns, certain media outlets and social media users have unknowingly facilitated the spread of false information by sharing unverified reports and sensationalist stories [63].

Victims: Individuals or groups who are negatively affected by disinformation and misinformation. They may include the general public, specific communities, or even individual targets. Victims suffer the consequences of false information, which can range from confusion and mistrust to tangible harm. During the COVID-19 pandemic, victims of misinformation included individuals who, believing false information about the virus or vaccines, engaged in risky behaviors or avoided vaccination. This not only put their own health at risk but also contributed to broader public health challenges [14, 100].

Campaigners: Individuals or groups who actively oppose the narratives pushed by mainstream sources or agents of disinformation. They work to challenge, debunk, or offer alternative perspectives to the dominant information streams. This opposition can involve activism, critical commentary, or advocacy for transparency and fact-checking. For instance, during various health crises, campaigners have played a vital role in challenging misinformation about diseases and their treatments by promoting evidence-based information and correcting public misconceptions. Their efforts are crucial in fostering a more informed public discourse and combating the effects of misinformation [9].

Objectives: Refer to the intentions of the agents behind disinformation campaigns. These can vary widely and include political gain, economic advantage, social disruption, or ideological influence. Understanding the objectives is crucial for developing strategies to counteract disinformation effectively. For example, during election periods, disinformation agents may aim to undermine confidence in the electoral process, discredit political opponents, or polarize the electorate to achieve specific political outcomes [3, 54].

Negative Effects: Negative consequences suffered by the victims of disinformation and misinformation. These can include confusion, mistrust, social unrest, and tangible harm. The effects of disinformation can be far-reaching, impacting individuals, communities, and societal structures. In the context of public health, misinformation about medical treatments or preventive measures can lead to widespread harm. For instance, during the COVID-19 pandemic, the spread of misinformation about the virus, its transmission, and the vaccines contributed to confusion, fear, and poor health outcomes [16].

Amplifiers: The mechanisms through which false information is spread more widely. This can include social media platforms, news outlets, or other communication channels that inadvertently or intentionally propagate disinformation or misinformation. They are essential in the virality of false information, as their reach and influence can exponentially increase the spread of inaccurate narratives. For example, algorithms on social media platforms that prioritize engaging content can inadvertently amplify disinformation, as sensationalist and misleading posts often receive more interactions and are thus more widely disseminated [22, 54].

Correctors: Individuals or organizations dedicated to identifying and debunking false information, such as fact-checking organizations, journalists, and informed citizens. Correctors work to provide accurate information, clarify misunderstandings, and reduce the impact of false narratives. During the COVID-19 pandemic, fact-checking organizations

and public health authorities actively worked to debunk myths and provide accurate information about the virus and vaccines [54].

Platforms: Refer to the digital and physical spaces where information is shared and consumed. These include social media networks, news websites, and traditional media outlets. The policies and algorithms of these platforms significantly influence the spread of disinformation and misinformation. Social media platforms, for example, can either facilitate the rapid dissemination of false information through viral sharing mechanisms or help curb it by implementing stringent content moderation policies and promoting verified information sources [35].

Algorithms: Systems used by social media platforms and search engines are instrumental in shaping the content users are exposed to. These mechanisms can unintentionally amplify disinformation and misinformation by favoring sensational, controversial, or highly engaging material, regardless of its factual accuracy. Efforts to address this issue focus on refining these systems to more effectively detect and downrank false information, while prioritizing verified and trustworthy sources [35, 90].

Bots and Trolls: Automated or human-operated accounts used to amplify false information. Bots can generate and spread vast amounts of content quickly, creating the illusion of widespread support or belief in false narratives. Trolls, on the other hand, engage in disruptive behavior by posting inflammatory or off-topic messages to provoke and derail discussions. These entities are often employed in coordinated disinformation campaigns to increase the reach and impact of false information [32].

In conclusion, understanding the various elements involved in information campaigns is essential for addressing the challenges posed by disinformation and misinformation. By identifying these key elements we can develop more effective strategies to combat the spread of false information. This comprehensive approach, combined with education, technological solutions, policy measures, and individual responsibility, can help protect the integrity of information in the digital age and foster a more informed and resilient society.

2.3. Conspiracy Theory Detection

Building on the discussion of disinformation and misinformation, conspiracy theories represent a potent subset of disinformation, characterized by false or unverified explanations for events, often attributing them to secret, malevolent groups or organizations. Their

detection is critical for maintaining societal trust and cohesion [28]. On the other hand, critical thinking, rooted in academic discourse, challenge traditional power structures and ideologies but are sometimes misinterpreted as conspiracy theories when they question established narratives. Several techniques are used to identify conspiracy theories. Textual analysis remains one of the most prominent, focusing on key terminologies and narratives that reflect unsupported claims, often framed in emotionally charged language. Natural language processing (NLP) tools facilitate large-scale detection by analyzing these patterns across diverse online content [62].

Research has also explored psycho-linguistic traits to detect conspiracy theory propagators, analyzing factors like sentiment and rhetorical style [34]. For example, an analysis of Reddit revealed that while conspiracy theorists made up only 5% of users, they were responsible for 64% of the comments [98]. Such patterns are useful in detecting individuals most likely to spread conspiratorial content. Taxonomies [49], like Holour's classification of "insiders" versus "outsiders" [42], further refine detection methods by highlighting in-group/out-group dynamics prevalent in conspiracy discourse. Emotional responses often differ between conspiracy and critical theory content. Conspiracy theories are more likely to evoke emotions like anger, disgust, or fear, while critical thinking may provoke reactions such as sadness or joy. Tools like Hurtlex[1] help analyze these emotional cues, providing insights into the engagement strategies employed in conspiracy-laden content [7].

Automated fact-checking tools are invaluable for verifying claims and countering conspiracy theories. They cross-reference content with verified databases, flagging discrepancies in real-time [37]. However, Brandolini's law highlights the challenge of refuting false claims, which often requires more effort than creating them [11]. User behavior analysis complements these methods by examining interactions with content, such as likes, shares, and comments, to identify patterns typical of conspiracy theorists and their networks.

To differentiate between conspiracy and critical thinking, it's essential to recognize the academic rigor behind critical discourse. critical thinking are often published in scholarly journals and backed by research. Verifying their legitimacy involves cross-referencing with academic sources like JSTOR[2] or Google Scholar[3] [92]. critical thinking also differ structurally and linguistically from conspiracy theories. It employs more complex academic language and methodological frameworks, supported by extensive references to scholarly work. Content analysis tools can help distinguish between these critiques and unsupported conspiracy narratives.

[1] https://github.com/valeriobasile/hurtlex
[2] https://www.jstor.org
[3] https://scholar.google.com

2.4. Binary Classification

Having explored the landscape of disinformation and misinformation, along with the key elements involved in their dissemination, we now shift our focus to the computational techniques employed to detect and categorize such content. Among these, binary classification plays a fundamental role, particularly in distinguishing between categories like disinformation and reliable information, or between conspiracy and critical thinking narratives.

Binary classification is a specialized form of text classification, which involves assigning text documents to one of two predefined categories or labels. It plays a vital role in various applications, such as spam detection, sentiment analysis (positive vs. negative), and language identification [46]. By classifying large volumes of unstructured text into two meaningful categories, binary classification enables more efficient information retrieval and content management [89].

Historically, binary classification was performed manually by domain experts, such as doctors or auditors, who classified data based on established criteria or patterns [45]. However, manual classification was labor-intensive and prone to human error or subjectivity. Early attempts to automate binary classification in the 1960s and 1970s relied on rule-based systems, where classification was driven by manually created rules, such as using specific thresholds to detect fraud [25]. While effective in some cases, these rule-based systems struggled with adaptability and required constant updates from experts whenever the data or classification criteria changed.

In the 1980s and 1990s, statistical approaches began to replace rule-based systems. Algorithms like Naive Bayes became popular due to their simplicity and relative effectiveness [66]. The development of techniques to represent data numerically, such as feature scaling and normalization, enhanced the performance of statistical models in binary classification.

The late 1990s and early 2000s saw the introduction of machine learning algorithms, such as support vector machines (SVM) [44], decision trees, and k-nearest neighbors (k-NN), which shifted the paradigm toward data-driven approaches. These algorithms allowed models to learn from data patterns rather than relying on fixed rules, marking a significant step toward scalable, automated binary classification.

By the 2010s, deep learning techniques emerged as a transformative force in binary classification. Unlike previous methods, deep learning models, especially convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks, automated the feature extraction process, learning representations of

data that captured both local and sequential dependencies [36]. The introduction of transformers [103], a neural architecture based solely on self-attention mechanisms, further revolutionized binary classification by enabling models to process large amounts of data without the limitations inherent to RNNs [99]. Transformer-based models have set new benchmarks in classification tasks. Models like BERT [27], GPT [107], RoBERTa [59], and T5 [70] have been fine-tuned for specific binary classification tasks with remarkable efficiency [103]. These models have been evaluated on benchmarks such as the GLUE dataset [102], achieving state-of-the-art results in tasks like sentiment analysis, natural language inference, and textual similarity.

Large Language Models (LLMs), such as OpenAI's GPT-4 [1], Meta's Llama 3 [29], and Google's Gemini 1.5 [93], have significantly advanced NLP by enabling sophisticated text understanding and generation. These models, trained on vast datasets, are commonly adapted for binary classification through fine-tuning and prompt engineering.

Fine-tuning tailors an LLM to a specific task by training it on labeled data, allowing it to capture task-relevant patterns while preserving its broad language capabilities. Studies have shown that fine-tuned LLMs frequently outperform traditional models on various binary classification benchmarks [105]. For example, Llama 3.1 has demonstrated superior performance in sentiment analysis tasks [68].

Prompt-based methods, on the other hand, leverage zero and few-shot learning capabilities of LLMs by framing classification tasks as text generation challenges. Here, prompts guide the model to generate a response representing the classification label, making this approach effective in scenarios with limited labeled data, achieving competitive results on sentiment analysis and topic classification tasks [57].

LLMs excel in tasks such as spam detection, fake news identification, and disinformation detection, leveraging contextual understanding to identify nuanced patterns. Their ability to detect complex patterns associated with disinformation and conspiracy theories makes them particularly suitable for these domains [101]. However, deploying LLMs for binary classification remains resource-intensive, often necessitating specialized hardware.

2.4.1. Ensembling Models in Binary Classification

Ensembling in binary classification is an indispensable technique for enhancing model performance, particularly in addressing the challenges posed by the high dimensionality and variability of textual data. By combining the predictions of multiple models, ensembles can significantly improve both accuracy and generalization, making them a powerful tool in binary natural language processing tasks. This improvement stems from the ability of

ensembles to aggregate the strengths of individual models, balancing out weaknesses and reducing the likelihood of overfitting to any particular aspect of the training data [108].

One common method in ensembling is Bagging (Bootstrap Aggregating), which trains multiple instances of the same model on different subsets of the training data, generated through bootstrapping. The individual models' predictions are then aggregated, typically via majority voting for binary classification tasks. Bagging reduces variance by ensuring that each model sees slightly different data, thereby leading to less correlated errors [12]. Random Forests, a popular extension of bagging applied to decision trees, have proven effective in binary classification by capturing a wide range of patterns from different perspectives within the data [13]. A visual representation of the process of Bagging can be seen in Figure 2.1.

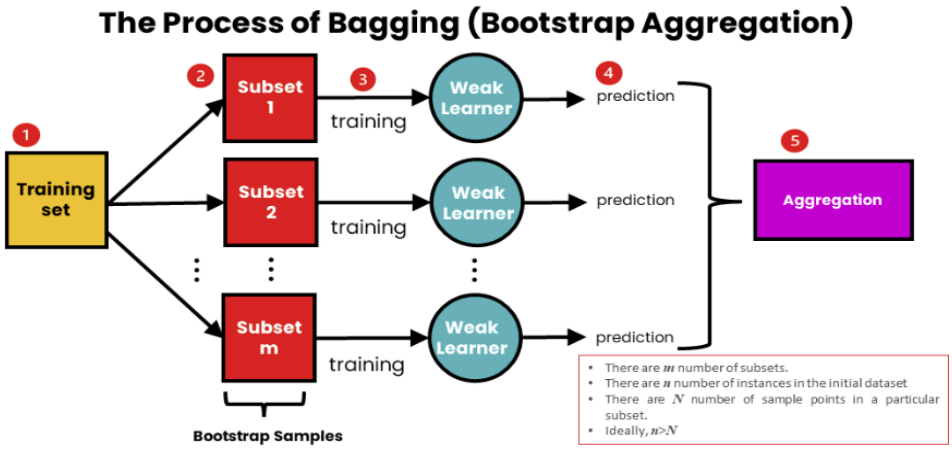


Figure 2.1: Process of Bagging: Boostrap Aggregation [71]

Boosting (Figure 2.2), in contrast to bagging, trains models sequentially, where each subsequent model focuses on correcting the errors made by the previous ones. This method emphasizes harder-to-classify instances, which makes boosting particularly adept at refining decision boundaries in noisy or complex datasets [33]. Algorithms like AdaBoost and Gradient Boosting Machines (GBM), including XGBoost [20] and LightGBM [47], are well-suited to binary classification tasks, where they excel at capturing intricate patterns in text data while being able to focus on difficult or borderline cases.

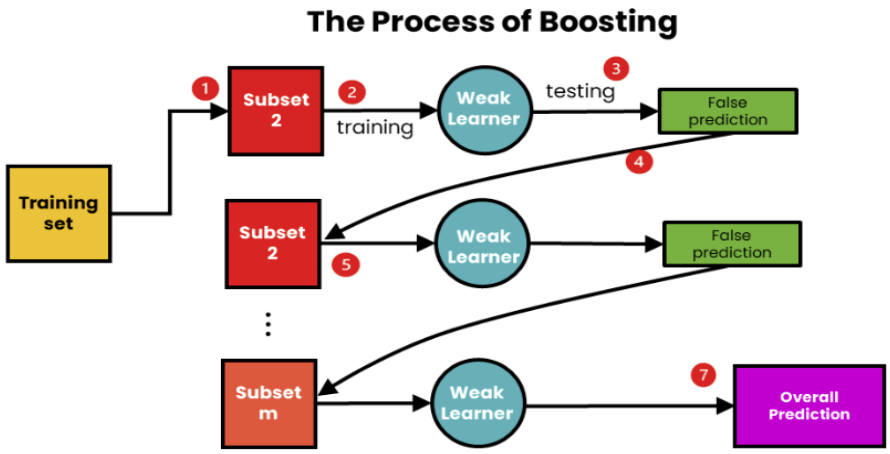


Figure 2.2: Process of Boosting [71]

Stacking, (Figure 2.3), is another advanced ensembling technique that combines predictions from several base models by training a meta-model to learn how to best integrate them [104]. In binary classification, this allows for the combination of diverse models, such as logistic regression, support vector machines (SVMs), and neural networks, to build an ensemble that captures different facets of the data. The meta-model learns to weight the outputs of these base models, leveraging their individual strengths to create a more robust and accurate overall classifier [108].

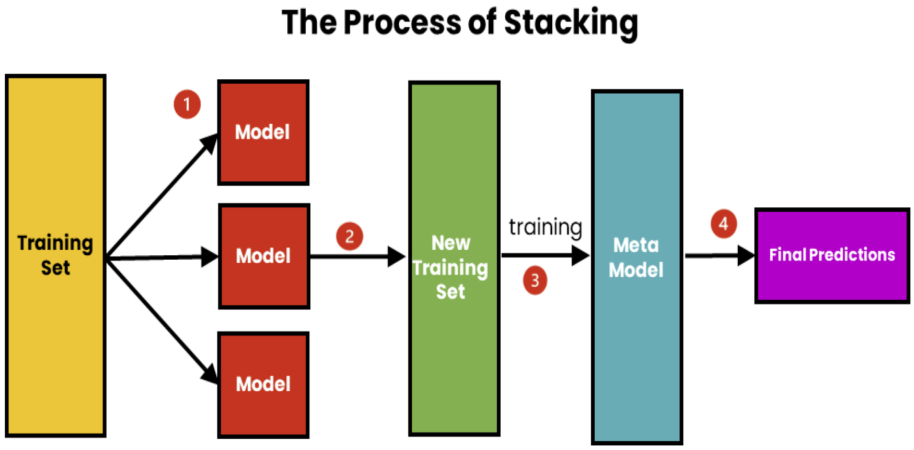


Figure 2.3: Process of Stacking [71]

A more straightforward approach to ensembling in binary classification involves voting classifiers, which combine predictions through Hard or Soft voting. Hard voting, as depicted in Figure 2.4, involves each base model casting a vote, with the final class being chosen based on the majority vote [50]. This method works better when the individual classifiers are relatively accurate and diverse, as it relies on the majority of models being correct.

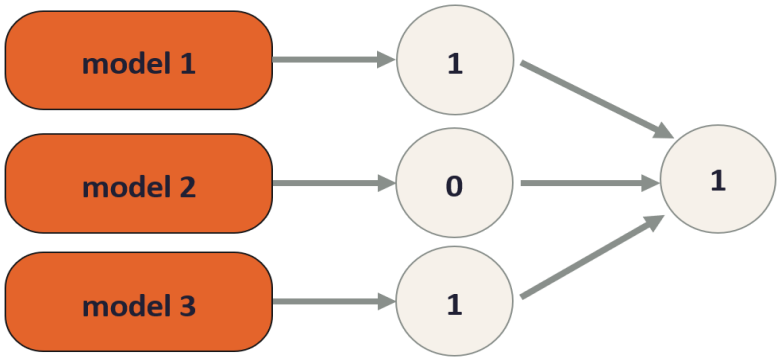


Figure 2.4: Hard Voting Ensemble

Soft voting, shown in Figure 2.5, takes a more refined approach by averaging the predicted probabilities of each class across the models. This method often performs better than hard voting, as it accounts for the confidence of the base models' predictions, giving more weight to models that are more certain of their outputs [108].

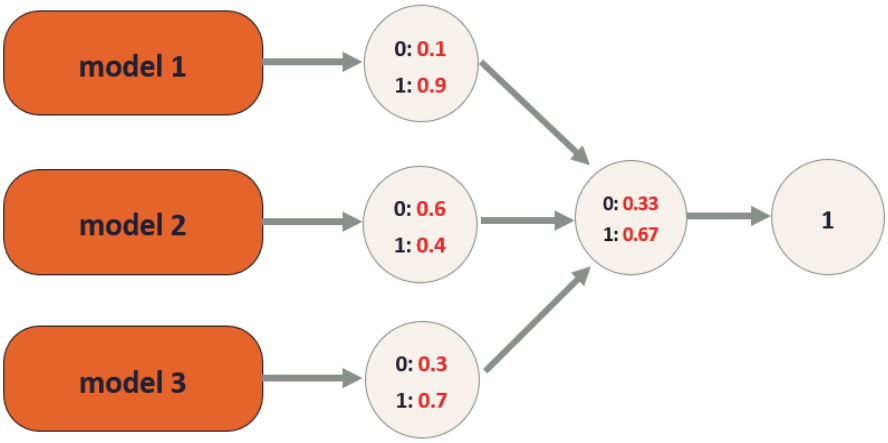


Figure 2.5: Soft Voting Ensemble

In practical terms, soft voting can be particularly useful in binary classification when base models vary significantly in their performance.

Despite its successes, binary classification still faces several ongoing challenges. Data quality is a persistent issue, as high-quality labeled data is essential for training robust models, yet acquiring such data at scale can be difficult. Imbalanced datasets, where one of the two classes is underrepresented, can also bias models, making it harder to achieve accurate predictions for the minority class [41]. Furthermore, deep learning models, particularly transformers, are often criticized for their lack of interpretability. As these models operate like "black boxes," it is crucial to develop methods for explaining their decisions, especially in sensitive applications like healthcare or finance [86]. Techniques like LIME [85]

and SHAP [60] are being explored to provide more transparency. Real-time processing is another challenge, as the volume of data requiring classification grows exponentially.

2.5. Token Classification

While binary classification focuses on categorizing entire documents or instances into one of two predefined classes, token classification delves deeper by assigning labels to individual tokens, words, subwords, or characters, within a text sequence. This token-level labeling enables more granular extraction of structured information from unstructured text, which is critical for Natural Language Processing tasks that require understanding the roles of each token in context. Token classification powers a variety of advanced NLP applications, including Named Entity Recognition (NER) [69], where entities such as names of people, organizations, or locations are identified; Part-of-Speech (POS) tagging [96], which assigns grammatical categories to words; and chunking [87], where sequences of words are grouped into meaningful units like noun or verb phrases.

Token classification provides a detailed view of text structure by labeling each token individually. This deeper level of analysis is crucial for more complex NLP tasks such as information retrieval [46], question answering [53], text summarization [58], and machine translation [6], where understanding the relationships between individual tokens and their contextual roles is predominant. For instance, NER, as shown in Figure 2.6, is a core application of token classification. In this task, specific tokens like names of people, organizations, or dates are identified within a sentence, adding structure and meaning to otherwise unstructured data [52].

Figure 2.6: Named Entity Recognition (NER) example

Historically, early approaches to token classification relied on manually crafted linguistic rules to assign labels to tokens [94]. These systems, however, had limited generalizability and scalability.

The shift to statistical models in the 1980s and 1990s marked a significant improvement. Methods such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) provided more robust solutions by modeling token relationships probabilistically. These models allowed systems to handle the ambiguity and variability of natural language more

effectively, as they could capture dependencies between tokens [51, 82].

As machine learning gained prominence in the late 1990s and early 2000s, token classification evolved further with algorithms such as decision trees, Support Vector Machines, and maximum entropy models [84]. However, these methods still relied heavily on manual feature engineering. Domain experts had to extract specific features from text, such as word frequencies or part-of-speech patterns, making the process labor-intensive.

The real breakthrough came in the 2010s with the rise of deep learning. Models like Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and later, transformers drastically changed token classification. These models automatically learned hierarchical representations of text and captured long-range dependencies between tokens, making them ideal for tasks such as NER, POS tagging, and chunking [91].

With the rise of deep learning, the success of token classification models depends not only on the architecture but also on the quality of the input text. Preprocessing, therefore, becomes a crucial step, as text needs to be prepared before it can be processed by machine learning models. This includes techniques like tokenization, which splits text into individual tokens; lemmatization, which reduces words to their root forms; and stop-word removal, which eliminates common but semantically insignificant words [46]. These preprocessing steps ensure that the input text is in a format suitable for analysis by token classification models.

One of the most revolutionary advancements in token classification has been the development of transformer-based models. As shown in Figure 2.7, transformers rely on self-attention mechanisms, allowing them to capture long-range dependencies between tokens more effectively than earlier methods like RNNs [99]. HuggingFace provides access to pre-trained transformer models like BERT [27], GPT [107], RoBERTa [59], and T5 [70], which have dramatically transformed token classification by allowing users to fine-tune these models for specific tasks with minimal computational resources [103]. For example, BERT has achieved state-of-the-art results on the CoNLL-2003 NER dataset [88], demonstrating the model's ability to understand contextual relationships at the token level. RoBERTa and T5 have also been fine-tuned for various token classification tasks, consistently outperforming traditional models.

The key innovation that transformer models bring is transfer learning, which allows pre-trained models to be fine-tuned on task-specific datasets, significantly reducing the training time and amount of labeled data required. Because of that, transformer models can be adapted to token classification tasks with minimal labeled data. This has lowered the barrier to entry for deploying state-of-the-art token classification systems, allowing

smaller organizations and even individual developers to leverage these advanced models. Additionally, multilingual models such as mBERT [76] and XLM-Roberta [23], are extending token classification capabilities across multiple languages. These models enable NLP systems to function in diverse linguistic environments, expanding their applicability globally.

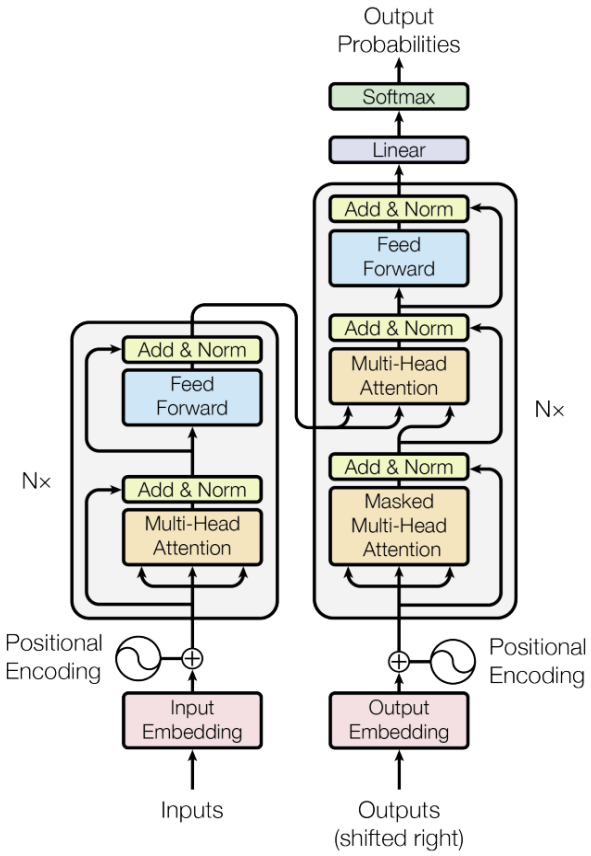


Figure 2.7: Transformer model architecture [99]

Large Language Models have further advanced token classification by enabling models to understand complex linguistic patterns at the token level. LLMs have been adapted for token classification tasks using approaches like fine-tuning and prompt engineering.

Fine-tuning LLMs for token classification involves modifying the model architecture to produce token-level outputs. This is achieved by adding a token classification head to the model, allowing it to assign labels to individual tokens in the input sequence.

Prompt-based methods for token classification utilize the generative capabilities of LLMs by crafting prompts that elicit token-level outputs. For instance, prompting the model to generate annotations or labels for each token in a sentence. This approach has been explored in few-shot and zero-shot settings, demonstrating the flexibility of LLMs in adapting to token classification tasks without extensive labeled data [38].

3 | Evolution of Disinformation Detection Tasks

The detection and analysis of disinformation and misinformation have evolved considerably in recent years, particularly as these phenomena have become more sophisticated and pervasive in digital media. Early research challenges, such as those presented in the *PAN*[1] and *MediaEval*[2] series, primarily centered on profiling users and detecting false information at a broad level. However, the need to understand the structure of disinformation narratives, especially those that blur the line between legitimate criticism and conspiracy theories, has led to a shift in focus. The *PAN 2024 Oppositional Thinking Analysis Challenge*[3] exemplifies this shift by introducing more complex tasks that emphasize the detection of the components and structures that define oppositional narratives.

3.1. The Evolution of Disinformation Detection Tasks

In the early stages of disinformation detection research, the primary goal was to identify users responsible for spreading fake news. The *PAN 2020*[4] task Profiling Fake News Spreaders on Twitter [83] was one of the pioneering efforts in this regard. It focused on profiling users by analyzing their linguistic and behavioral patterns to distinguish fake news spreaders from those sharing legitimate content. This task laid the groundwork for subsequent research by providing insights into how user behaviors could reveal patterns of disinformation dissemination.

In parallel to *PAN*, the *MediaEval* series contributed significantly to advancing the detection of disinformation through tasks that analyzed both textual content and social network structures. The *MediaEval 2020 Fake News: Corona Virus and 5G Conspiracy Multimedia Analysis Task*[5] [77] featured two primary subtasks: one focusing on natural

[1] https://pan.webis.de
[2] https://multimediaeval.github.io
[3] https://pan.webis.de/clef24/pan24-web/oppositional-thinking-analysis.html
[4] https://pan.webis.de/clef20/pan20-web/author-profiling.html
[5] https://multimediaeval.github.io/editions/2020/tasks/fakenews/

language processing to classify tweets related to COVID-19 and 5G, and the other on structure-based analysis using Twitter network data. The dataset included two sets of tweets that mentioned the topics of Coronavirus and 5G, comprising text, reposting time patterns, and basic information about the users who reposted them. Additionally, the dataset provided a Twitter follower network, detailing the Twitter users who shared each respective tweet. This design emphasized the importance of combining linguistic features with reposting behaviors and network-based detection methods.

Subsequent editions, *MediaEval 2021*[6] and *2022*[7], continued to emphasize the interplay between textual content and Twitter graph-based information for disinformation detection [78, 79]. These tasks underscored the necessity of understanding how disinformation narratives spread and evolve within social networks, focusing on the amplification of false content through structural and reposting patterns.

3.2. The PAN 2024 Oppositional Thinking Challenge

The *PAN 2024 Oppositional Thinking Challenge* represents a significant leap forward in the analysis of disinformation by addressing the complexities of oppositional narratives. Moving beyond detecting false information or profiling fake news spreaders, the challenge introduces a more sophisticated set of tasks focused on the structure of narratives themselves. The challenge is divided into two subtasks: (i) Distinguishing between critical and conspiracy texts, and (ii) Detecting elements of the oppositional narratives.

3.2.1. Binary Classification of Oppositional Narratives

The first task is a *binary classification task*, where the goal is to classify a given text as either reflecting *conspiratorial intent* or *critical thinking*. This task addresses the challenge of distinguishing between two types of oppositional thought that, on the surface, can appear quite similar. Conspiratorial narratives often mimic the language and rhetorical strategies of critical thinking, using factual information mixed with distortions or falsehoods to create plausible but misleading arguments. Critical thinking, on the other hand, is characterized by a reasoned approach that challenges dominant narratives based on evidence and logical analysis.

[6] `https://multimediaeval.github.io/editions/2021/tasks/fakenews/`
[7] `https://multimediaeval.github.io/editions/2022/tasks/fakenews/`

3.2.2. Detecting Elements of Oppositional Narratives

The second task, *detecting the elements of oppositional narratives*, is by far the most complex and significant aspect of the PAN 2024 challenge. Participants were required to identify and classify specific narrative elements within a text. These elements [49], *Agents*, *Facilitators*, *Victims*, *Campaigners*, *Objectives*, and *Negative Effects*, are essential for understanding the deeper structure of how narratives are framed and presented.

Agents refer to the individuals or entities responsible for the actions or outcomes described in the narrative. **Facilitators** are those who assist the agents, enabling the events to unfold. **Victims** are the individuals or groups who suffer the consequences of the actions taken by the agents, while **Campaigners** represent those who oppose or resist the mainstream narrative or actions of the agents. **Objectives** outline the intentions or goals of the agents, and **Negative Effects** capture the consequences or harm caused by the events described.

The challenge of this task lies in the fact that these narrative elements are often embedded within one another, allowing for spans that encompass multiple roles. For example, a span-text discussing a *Negative Effect* may include references to the *Victim* affected by it. This overlapping nature requires an analysis approach that can detect and account for the fluid and nested relationships between these roles within oppositional narratives.

This effort reflects the complexity of real-world disinformation and conspiracy theories. These narratives are often crafted to blur the lines between fact and fiction, and their effectiveness lies in their ability to weave multiple narrative elements together to create a compelling, although false, story. Understanding how these narratives are structured, and how different elements interact, is crucial for developing tools that can effectively combat disinformation.

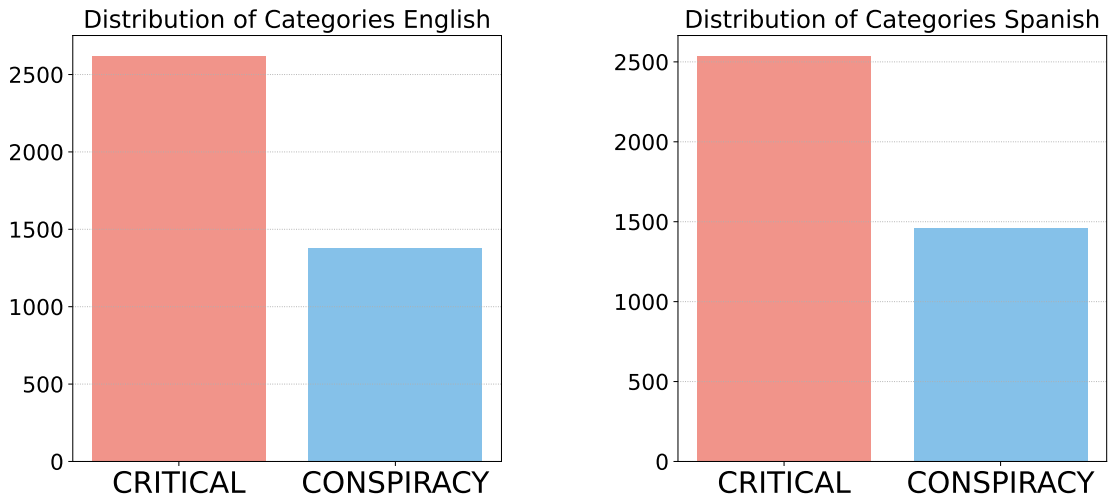
This task also opens the door to more advanced forms of narrative analysis, such as understanding how different elements contribute to the overall persuasiveness of a narrative. By identifying patterns in the use of *Agents*, *Facilitators*, and the other elements, researchers can begin to map out common strategies used in conspiracy theories and in critical thinking. This could lead to more effective countermeasures, such as targeted fact-checking or interventions aimed at breaking down specific narrative structures.

4 | Dataset Description

In this chapter, we will analyze the dataset provided, examining its structure, the distribution of binary categories and the span-level annotations that highlight key elements within oppositional narratives.

4.1. Dataset Overview and Binary Categories

For this task, the organisers provided a JSON file containing all texts in the training dataset along with their annotations. Each text is represented by a dictionary that includes the ID, tokenized text, binary category, and span annotations with their categories. Span annotations are provided as a list of dictionaries, with each dictionary representing an annotated span and detailing the span's category and text, specified by the start and end characters. The training dataset, comprising 4000 records extracted from Telegram comments, was released with all annotations, while the test dataset, consisting of 1000 records, was released only with "id" and the "text" field [9, 48].



(a) Binary categories for the English dataset (b) binary categories for the Spanish dataset

Figure 4.1: Distribution of binary categories for English and Spanish train datasets

Label	English	(%)	Spanish	(%)
CRITICAL	2621	65.53%	2538	63.45%
CONSPIRACY	1379	34.48%	1462	36.55%

Table 4.1: Distribution of binary labels for the English and Spanish train datasets

As summarized in Table 4.1 and shown in Figures 4.1a and 4.1b, the binary classification task for distinguishing critical from conspiracy texts reveals an inherent class imbalance in both the English and Spanish datasets. The English dataset has 65.53% critical texts compared to 34.48% conspiracy texts, while the Spanish dataset shows a similar trend with 63.45% critical and 36.55% conspiracy texts. This imbalance poses a significant challenge for model training, as models may become biased towards the more frequent class (critical texts), potentially overlooking subtle features of conspiracy texts. The similarity in the proportions of critical and conspiracy texts across both languages suggested that the models trained on one dataset might be adaptable to the other with minimal adjustments. That was indeed the approach that we adopted for the shared task. Examples for each category are shown in Table 4.2 and in Table 4.3.

Text	Category
CNN continued its wall to wall broadcasts calling for unvaccinated people to be punished, with analysts again calling for those who have not gotten the COVID shots to be segregated from society and forced to pay for tests every single day. https://summit.news/2021/07/23/cnn ... /	CRITICAL
Are women suffering adverse injuries and deaths from the Covid - 19 vaccinations in far greater numbers than men? Data from the EU suggests that this is indeed the case with women reporting more than three times the issues that men are. This data is not provided by the MHRA in the UK but you can access the EU data here by vaccine manufacturer https:// expose - news .com/2022/06/08/ ...	CRITICAL
Joe Biden's Commerce Secretary has claimed that nobody is being forced to get vaccinated, despite last week's announcement that millions of Americans will be mandated to take the shot in order to go to work . https :// summit . news / 2021 / 09 / 15 / ...	CRITICAL

Table 4.2: Examples of critical texts

Text	Category
Agenda 21 ... Goal 1: End poverty in all its forms everywhere Translation: Centralized banks, IMF, World Bank, Fed to control all finances , digital one world currency in a cashless society Goal 2: End hunger, achieve food security: GMO Goal 3: Ensure healthy lives and promote well - being for all at all ages Translation: Mass vaccination, Codex Alimentarius ... Population control through forced "Family Planning" ...	CONSPIRACY
THE NEW WORLD ORDER JUST PUSHED THE TURBO BUTTON - - WHO STILL THINKS IT'S A CONSPIRACY THEORY? Think back to WWII. This is the same thing. They cause a big global crisis to enact their agenda through treaties and create hegemony. Why people can not see this as the same playbook is just mind blowing to me. All wars are literally cover for these agendas. Covid is WWIII without the guns and bombs. https://www.dailymail.co.uk/ ... @DismantlingTheCabal	CONSPIRACY
PfizerGate Scandal: The Worldwide Cover-up of Data to disguise the fact Covid-19 Vaccines cause VAIDS. Health authorities around the world are manipulating figures in an attempt to hide from the general public that the Covid-19 injections are causing the fully vaccinated to develop Vaccine Acquired Immune Deficiency Syndrome; and we can prove it . . .	CONSPIRACY

Table 4.3: Examples of conspiracy texts

4.2. Span-Level Annotations and Analysis

Apart from the binary label, the messages also include annotated token-level elements. According to the dataset's authors [49], these elements represent "the key elements of oppositional narratives." The six span categories defined by the authors of the PAN task are:

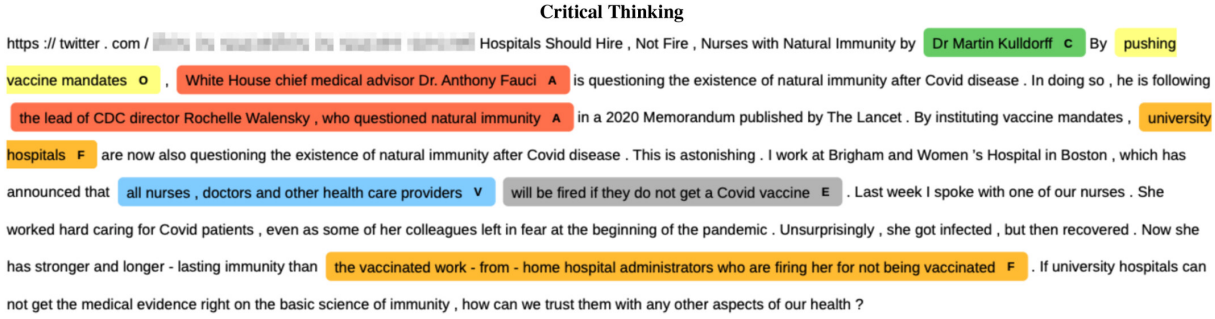
- **Agents**: Individuals or entities responsible for the actions and/or negative effects.
- **Facilitators**: Those who assist the agents.
- **Victims**: Those who suffer the consequences of the agents' actions.
- **Campaigners**: Those who oppose the mainstream narrative.
- **Objectives**: The goals or intentions of the agents.
- **Negative effects**: The adverse consequences experienced by the victims.

Table 4.4 shows that a single text message can contain multiple span elements of the same category. It is also important to note that some texts may not contain any span elements. Additionally the span text can partially overlap and this increases exponentially the difficulty in finding all the span elements.

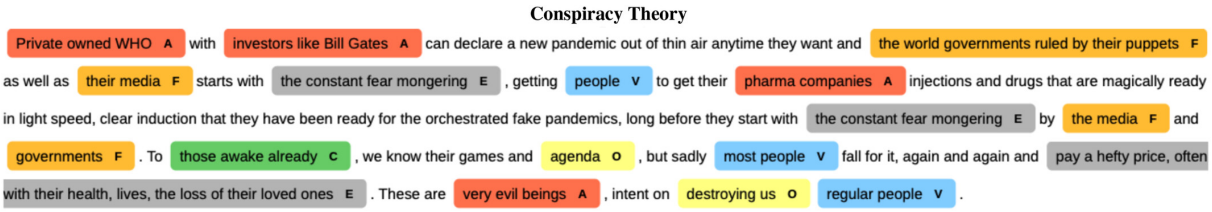
Span Text	**Category**
Joe Biden's Commerce Secretary	AGENT
THE NEW WORLD ORDER	AGENT
vaccine manufacturer	AGENT
analysts	FACILITATOR
CNN	FACILITATOR
Health authorities around the world	FACILITATOR
millions of Americans	VICTIM
unvaccinated people	VICTIM
women	VICTIM
summit	CAMPAIGNER
expose-news	CAMPAIGNER
DismantlingTheCabal	CAMPAIGNER
Centralized banks, IMF, World Bank, Fed to control all finances, digital one world currency in a cashless society	OBJECTIVE
Mass vaccination, Codex Alimentarius Goal	OBJECTIVE
Population control through forced "Family Planning"	OBJECTIVE
the fact Covid - 19 Vaccines cause VAIDS	NEGATIVE_EFFECT
women reporting more than three times the issues that men are	NEGATIVE_EFFECT
adverse injuries and deaths from the Covid - 19 vaccinations	NEGATIVE_EFFECT

Table 4.4: Span elements from the critical texts in Table 4.2 and from the conspiracy texts in Table 4.3 used for the span-level detection task

In order to better understand the structure of the oppositional narratives scheme, a visual example can be found in Figure 4.2. In Figure 4.2a we can observe a critical thinking message annotated with the six different categories, while in Figure 4.2b we can observe a conspiracy theory message.



(a) A critical message annotated with elements of oppositional narrative

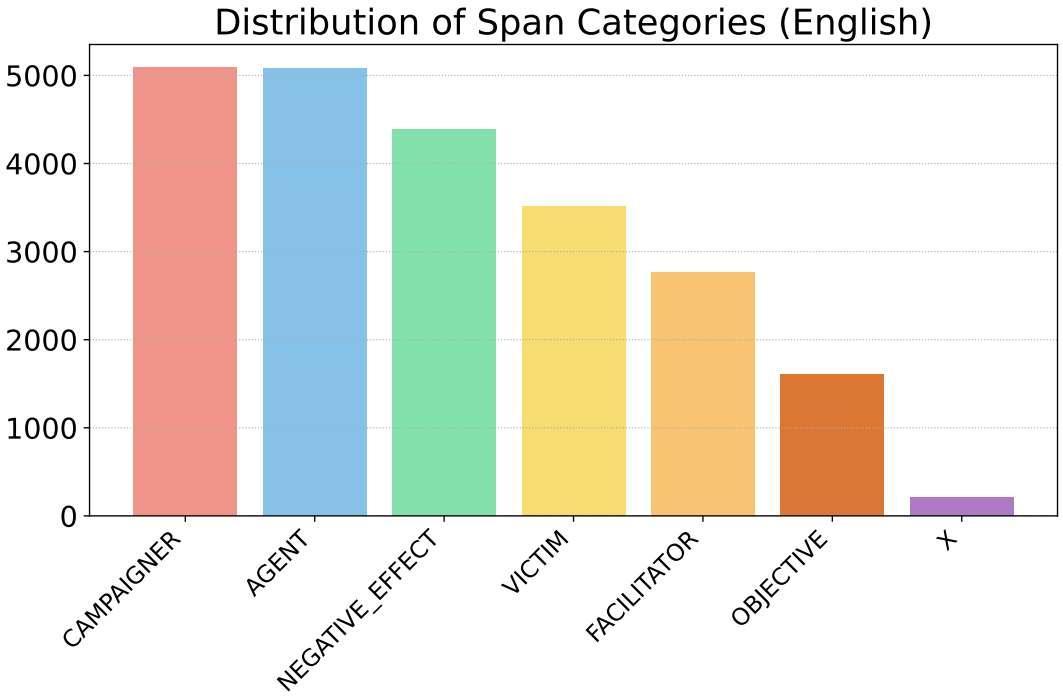


(b) A conspiracy message annotated with elements of oppositional narrative

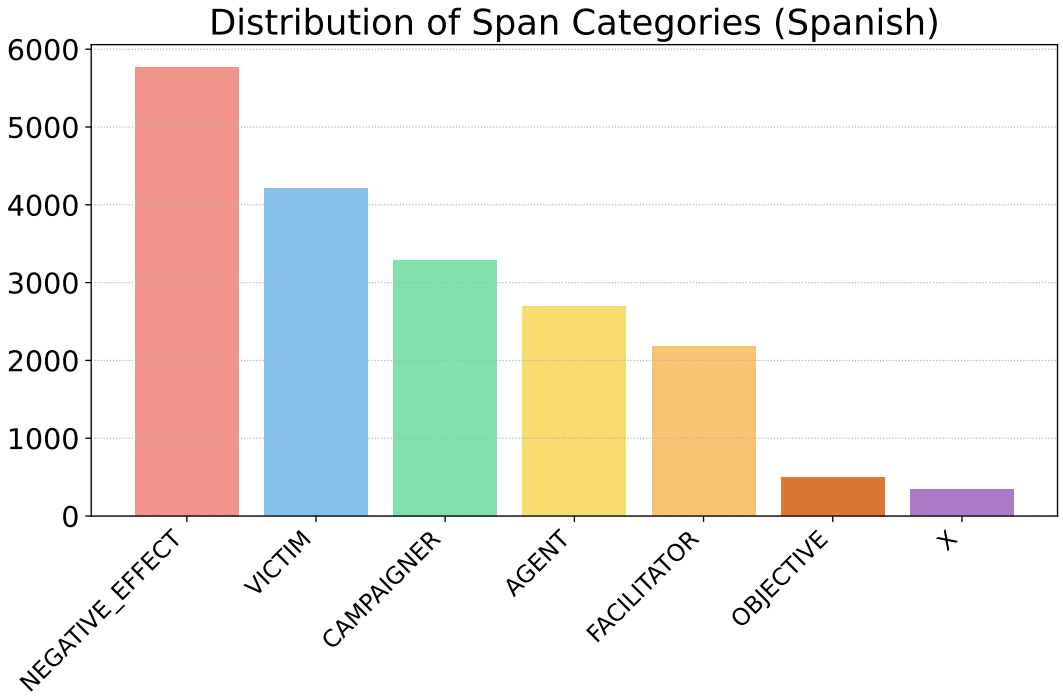
Figure 4.2: A conspiracy and a critical message annotated with elements of oppositional narrative: Agents (A), Facilitators (F), Campaigners (C), Victims (V), Objectives (O) and Negative Effects (E) [49]

4.2.1. Distribution of Span Categories

As shown in Figures 4.3a and 4.3b and in Table 4.5, the distribution of key elements within oppositional narratives varies significantly between the English and Spanish datasets, indicating potential differences in narrative focus.



(a) Distribution of span text categories for the English dataset



(b) Distribution of span text categories for the Spanish dataset

Figure 4.3: Distribution of span text categories for English and Spanish train datasets. The label "x" represents the texts where no label appears for the task.

Label	English	(%)	Spanish	(%)
CAMPAIGNER	5096	22.70%	3285	17.63%
AGENT	5082	22.63%	2698	14.47%
NEGATIVE_EFFECT	4387	19.54%	5770	30.96%
VICTIM	3517	15.67%	4213	22.61%
FACILITATOR	2763	12.31%	2174	11.67%
OBJECTIVE	1602	7.14%	493	2.65%

Table 4.5: Distribution of span text labels for English Spanish train datasets

In the English dataset, the categories of CAMPAIGNER and AGENT are more prominently represented, with counts of 5096 and 5082, respectively. This suggests that English texts may place a stronger emphasis on identifying and challenging the actors involved in conspiracies. In contrast, the Spanish dataset has a significantly higher count of NEGATIVE_EFFECT span texts (5770), indicating a focus on the consequences and impacts of the events described.

The notable differences in span distributions between the English and Spanish datasets underscore the importance of understanding cultural and linguistic nuances in conspiracy narratives. The English dataset's broader focus on actors and their goals might reflect different narrative styles or emphases compared to the Spanish dataset, which concentrates more on the effects and victims. These differences could stem from varying annotation guidelines or cultural perspectives on conspiracy theories.

4.2.2. In-Depth Analysis of Span-Level Labels in English and Spanish on the Train Datasets

Figure 4.4 provides a visual overview of the distribution, while Table 4.6 gives a detailed breakdown of these categories. The contrast between the CONSPIRACY and CRITICAL texts highlights distinctive narrative strategies and focuses.

Examining the distribution of span-level labels in conspiracy and critical texts within the English dataset offers intriguing insights into how these narratives are framed and what elements are emphasized.

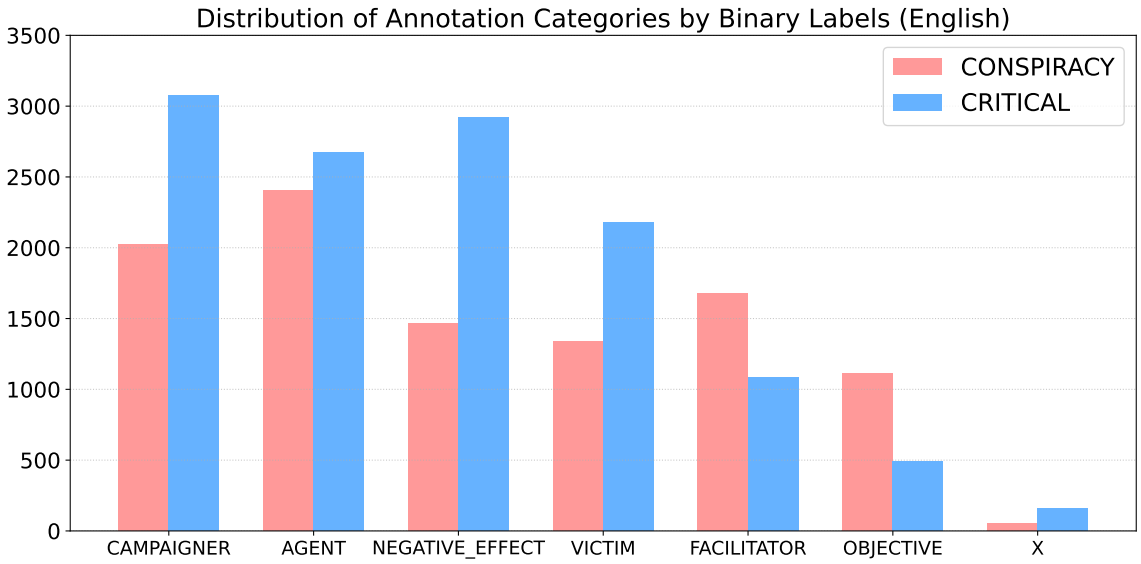


Figure 4.4: Distribution of categories by binary labels in English

Span Category	CONSPIRACY	(%)	CRITICAL	(%)
NEGATIVE_EFFECT	1465	33.39%	2922	66.61%
AGENT	2406	47.34%	2676	52.66%
CAMPAIGNER	2023	39.68%	3073	60.32%
VICTIM	1341	38.13%	2176	61.87%
FACILITATOR	1677	60.70%	1086	39.30%
OBJECTIVE	1110	69.29%	492	30.71%
X	50	24.27%	156	75.73%

Table 4.6: Distribution of categories by binary labels in English

Starting with the NEGATIVE_EFFECT category, critical texts exhibit a significantly higher frequency of annotations, with 2922 instances (66.61%) compared to 1465 instances (33.39%) in conspiracy texts. This disparity suggests that critical narratives place a greater emphasis on the negative repercussions of the events being discussed, highlighting the adverse societal or individual impacts of conspiracies. The high volume of annotations in this category indicates that critical texts are particularly concerned with illustrating the harmful impacts of the events or decisions, which aligns with a narrative strategy focused on the broader consequences of these phenomena.

In contrast, the AGENT category is notably prominent in conspiracy texts, accounting for 2406 annotations (47.34%), and remains significant in critical texts with 2676 annotations (52.66%). This suggests that conspiracy narratives are heavily focused on identifying and

scrutinizing the individuals or entities driving the conspiracies. The higher percentage in conspiracy texts underscores a narrative strategy aimed at pinpointing and examining the key players involved, reflecting a desire to assign responsibility and understand the roles of specific actors in these narratives.

The CAMPAIGNER category also exhibits an important distribution. Conspiracy texts have 2023 annotations (39.68%), whereas critical texts have 3073 annotations (60.32%). The higher percentage in critical texts suggests that critical narratives focus more on individuals actively challenging mainstream narratives, emphasizing the opposition to conspiratorial ideas rather than promoting them. This dynamic highlights how critical texts are more concerned with understanding and addressing the resistance to conspiracies rather than propagating them.

In contrast, the VICTIM category presents 1341 annotations (38.13%) in conspiracy texts and 2176 annotations (61.87%) in critical texts. The more significant representation of victims in critical texts points to an in-depth exploration of the harm caused by conspiracies, likely emphasizing the societal and personal impacts more critically than conspiracy texts, which may downplay the suffering caused by these narratives.

The FACILITATOR category is particularly striking in conspiracy texts, having 1677 annotations (60.70%), compared to the 1086 annotations (39.30%) in critical texts. This suggests that conspiracy narratives place considerable focus on those who enable or support the conspiratorial activities. The higher proportion in conspiracy texts highlights a narrative concern with understanding the support structures and individuals who assist in the execution of conspiracies, pointing to a more complex view of how conspiracies are carried out and sustained.

Finally, the OBJECTIVE annotations are most prevalent in conspiracy texts, with 1110 annotations (69.29%), compared to the 492 annotations (30.71%) contained in critical texts. This significant difference implies that conspiracy narratives are particularly concerned with the underlying goals and aims of the conspiratorial activities. The focus on OBJECTIVE reflects an interest in understanding the motivations and purposes behind the conspiracies, contrasting with critical texts that are less concerned with these aspects and more focused on immediate impacts.

These span-level label distributions indicate distinct narrative approaches in English texts. Conspiracy narratives focus heavily on agents and facilitators, aiming to dissect the individuals responsible and their intentions. This suggests a more investigative tone, as these texts delve into who is behind the conspiracies and what they aim to achieve. Critical texts, on the other hand, place a greater emphasis on the negative effects and victims,

underscoring the damage caused by conspiracies. This divergence highlights two primary focuses: while conspiracy narratives are interested in unraveling the "who" and "why," critical texts are more concerned with examining the "what", the real-world consequences of these actions.

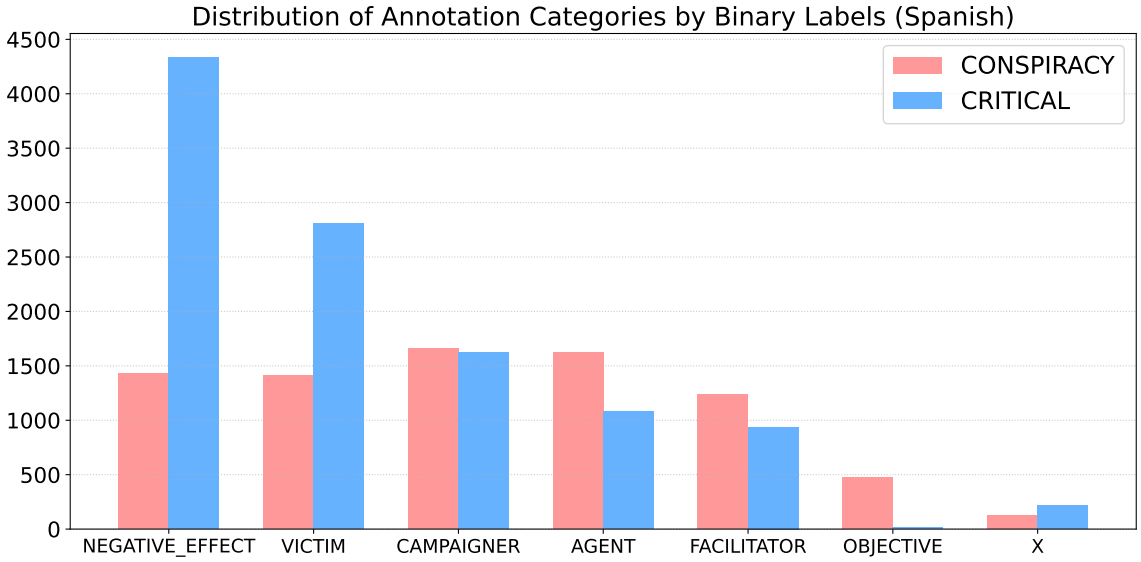


Figure 4.5: Distribution of categories by binary labels in Spanish

Span Category	CONSPIRACY	(%)	CRITICAL	(%)
NEGATIVE_EFFECT	1433	24.83%	4337	75.17%
AGENT	1621	60.07%	1077	39.93%
CAMPAIGNER	1662	50.59%	1623	49.41%
VICTIM	1408	33.42%	2805	66.58%
FACILITATOR	1236	56.87%	938	43.13%
OBJECTIVE	475	96.34%	18	3.66%
X	123	36.18%	217	63.82%

Table 4.7: Distribution of categories by binary labels in Spanish

In the case of the Spanish dataset, the span-level label distribution is illustrated in Figure 4.5 and a detailed numerical breakdown is provided in Table 4.7.

In the NEGATIVE_EFFECT category, there is a marked difference between the two types of texts. Critical texts overwhelmingly dominate this category, with 4337 annotations (75.17%) compared to just 1433 annotations (24.83%) in conspiracy texts. This suggests that critical texts in Spanish are more focused on illustrating the harmful consequences of the events described. This narrative approach emphasizes the severity and broader

impacts of the issues, reflecting a strategy that highlights the negative repercussions for individuals and society.

On the other hand, the AGENT category shows a stronger presence in conspiracy texts, accounting for 1621 annotations (60.07%) versus 1077 annotations (39.93%) in critical texts. This indicates that Spanish conspiracy narratives place a significant emphasis on identifying and scrutinizing the key individuals or entities behind the conspiracies. This focus suggests a narrative strategy aimed at understanding and exposing the orchestrators of these conspiracies.

The CAMPAIGNER category is almost evenly split between conspiracy (50.59%) and critical (49.41%) texts, respectively of 1662 and 1623 annotations. The similar distribution suggests that both types of texts focus equally on those challenging conventional views.

When it comes to the VICTIM category, critical texts again show a notable emphasis, with 2805 annotations (66.58%) compared to 1408 annotations (33.42%) in conspiracy texts. This suggests that Spanish critical texts focus more on the impact and experiences of those affected by conspiracies, highlighting the personal and societal consequences in their narratives.

The FACILITATOR category is more prevalent in conspiracy texts, with 1236 annotations (56.87%), than in critical texts, were is represented by 938 annotations (43.13%). This suggests that conspiracy narratives have a stronger focus on those who enable or support the conspiracies, providing insight into the support structures and individuals involved in these activities.

Although the OBJECTIVE category shows a striking disparity, with conspiracy texts dominating at 475 annotations (96.34%) compared to only 18 annotations (3.66%) in critical texts, the overall number of OBJECTIVE span-texts is relatively low. This means that, despite the high percentage within conspiracy texts, the absolute presence of this category is not particularly significant.

Overall, the distribution of span-level labels in the Spanish dataset reveals distinct narrative patterns compared to the English dataset. Spanish conspiracy texts emphasize key players, focusing heavily on the agents' actions and their aims. Critical texts, by contrast, highlight the negative effects and victims, placing more emphasis on the outcomes of conspiratorial actions. This suggests that Spanish narratives approach conspiracies with a different lens, concentrating on the motivations and actions of the key figures in conspiracy narratives, while critical texts focus on the harm inflicted by these actions. These differences may stem from cultural or linguistic variations in how conspiracy and critical

narratives are framed, providing insight into how Spanish texts conceptualize the causes and effects of conspiratorial thinking. Understanding these narrative strategies allows for a clearer view of how conspiracies and critical issues are discussed in Spanish contexts.

4.2.3. Comparative Analysis Across English and Spanish Datasets

Figure 4.6 presents a comparative analysis of span-level categories across conspiracy and critical texts in both English and Spanish datasets, merging the two figures previously seen. The dotted lines represent the Spanish dataset while the bars with solid colors represent the English dataset.

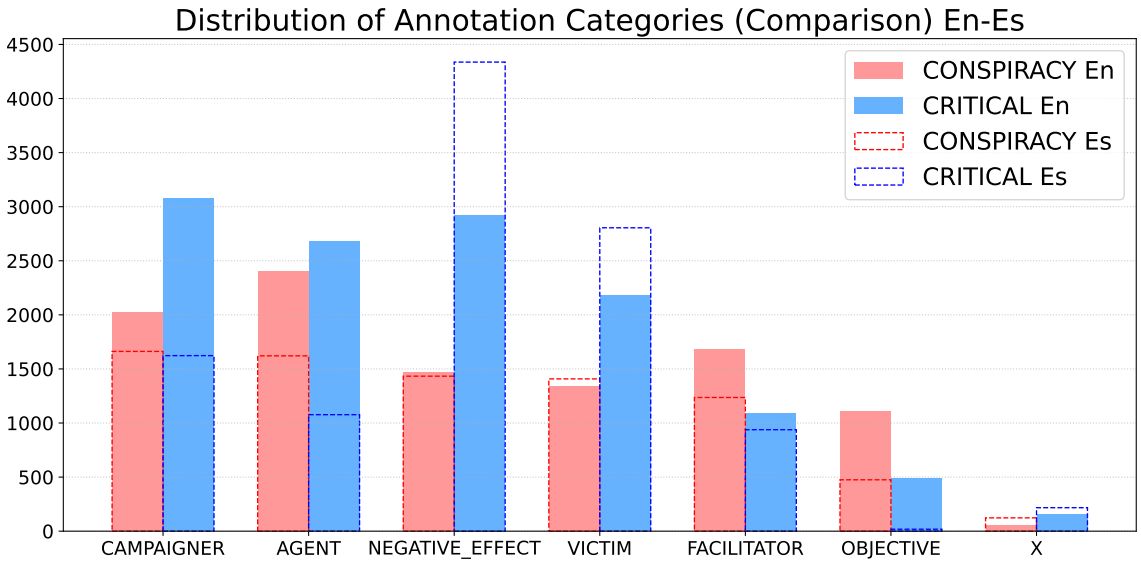


Figure 4.6: Distribution of annotation categories: comparison

In general, the Spanish dataset shows a pronounced disparity between conspiracy and critical texts, especially in categories such as NEGATIVE_EFFECT and VICTIM, where critical texts have significantly higher counts. This suggests a stronger emphasis on the adverse impacts of conspiracies and the experiences of those affected. However, the difference in NEGATIVE_EFFECT between Spanish and English critical texts is not as stark, as English critical texts also have substantial counts in this category.

Conversely, while Spanish conspiracy texts place a strong focus on AGENTS and FACILITATORS, this emphasis is even greater in English conspiracy texts, which show higher counts for both categories. This balance in English conspiracy texts contrasts with the suggestion that Spanish texts place a deeper emphasis on these categories.

In the English dataset, CAMPAIGNER and OBJECTIVE annotations reveal distinct

patterns. English critical texts have a notably higher count of campaigner annotations, indicating a strong focus on those opposing conspiracies. This emphasis is also present in Spanish texts, though the distribution between conspiracy and critical texts is more balanced. In terms of OBJECTIVES, English conspiracy texts show a stronger emphasis compared to critical texts, while in the Spanish dataset, OBJECTIVES are largely confined to conspiracy texts.

4.2.4. Comparative Insights: English vs. Spanish Text Lengths

Taking a further step, firstly we compared the lengths of the texts, as detailed in Table 4.8. This comparison reveals that, on average, conspiratorial texts are nearly twice as long as critical texts. Specifically, English conspiratorial texts have an average length of 742.9 characters, while English critical texts average 476 characters. Similarly, Spanish conspiratorial texts average 1112 characters, whereas Spanish critical texts average 641.2 characters. This suggests that conspiratorial texts tend to be more elaborate and possibly contain more detailed arguments or narratives compared to critical texts. Additionally, in the English dataset, each text message has an average of 5.61 span annotations, while the Spanish dataset has an average of 4.66 span elements per message.

Second, we observed the standard deviation of text lengths, which indicates the variability within each category. For English texts, the standard deviation for conspiratorial texts is 740.2, compared to 479.3 for critical texts. For Spanish texts, the standard deviation for conspiratorial texts is 945.3, compared to 577.8 for critical texts. The higher standard deviation in conspiratorial texts implies greater variability in their lengths, suggesting that some conspiratorial texts are much longer or shorter than others.

Additionally, we looked at the minimum and maximum lengths of texts in each category. The shortest English critical text has 78 characters, while the longest has 4695 characters. In contrast, the shortest English conspiratorial text has 88 characters, and the longest has 4346 characters. For Spanish texts, the shortest critical text has 99 characters, and the longest has 4761 characters, whereas the shortest conspiratorial text has 123 characters, and the longest has 4313 characters. This range of lengths further highlights the tendency for conspiratorial texts to include more content, which could be due to the complexity or thoroughness of the narratives they present.

These findings are summarized in Table 4.8 below, which provides a detailed statistical overview of text lengths across different categories and languages:

	Dataset	Count	Mean	Std Deviation	Min	Max
CRITICAL	English	2621	476	479.3	78	4695
	Spanish	2538	641.2	577.8	99	4761
CONSPIRACY	English	1379	742.9	740.2	88	4346
	Spanish	1462	1112	945.3	123	4313

Table 4.8: Statistics of text lengths (expressed in characters) by category

The second analysis investigated the relationship between each span element and the span categories. This was done to explore the potential connection between critical and conspiracy narratives and their respective narrative elements.

The analysis involved counting the number of span elements in each text and calculating the point-biserial correlation coefficient [15] between these counts and the category of the span. Here, the gold label is treated as a binary value: zero for critical and one for conspiracy. Table 4.9 displays the correlation between each annotation and the category label.

Two methods were used to count the annotations. The first method used an integer to represent the number of times each annotation appeared in the text message. The second method used a boolean to indicate whether at least one annotation of a certain type was present in the text, which was then used to calculate the Phi coefficient [30]. This analysis was conducted for both datasets. Generally, the boolean values showed a stronger correlation with the gold label than the count values. A stronger correlation suggests that certain narrative elements are more prevalent in texts labeled as conspiracy. Conversely, a negative correlation indicates that an element is more common in texts labeled as critical.

		AGENT	FACILIT.	VICTIM	CAMPAIG.	OBJEC.	NEG. EFF
English	Mult	0.11	0.17	0.04	0.07	0.30	-0.013
	Bin	0.18	0.17	-0.01	-0.05	**0.35**	-0.07
Spanish	Mult	0.18	0.17	-0.04	0.12	0.32	-0.11
	Bin	0.17	0.12	0.12	-0.10	**0.37**	-0.23

Table 4.9: Pearson coefficient correlation for the appearance of annotations in a text with the gold label category

In Figure 4.7, we examine the correlations among the binary key element indicators in the English dataset. The observed mild correlations suggest that the pairs of *Agent-Objective*, *Victim-Negative Effect*, and *Facilitator-Objective* frequently appear together.

This pattern aligns with the nature of conspiracy narratives, where an *Agent* pursues an Objective with the assistance of *Facilitators*, which subsequently results in *Negative Effects* for the *Victims*. Additional analysis of the dataset can be found in Appendix A.

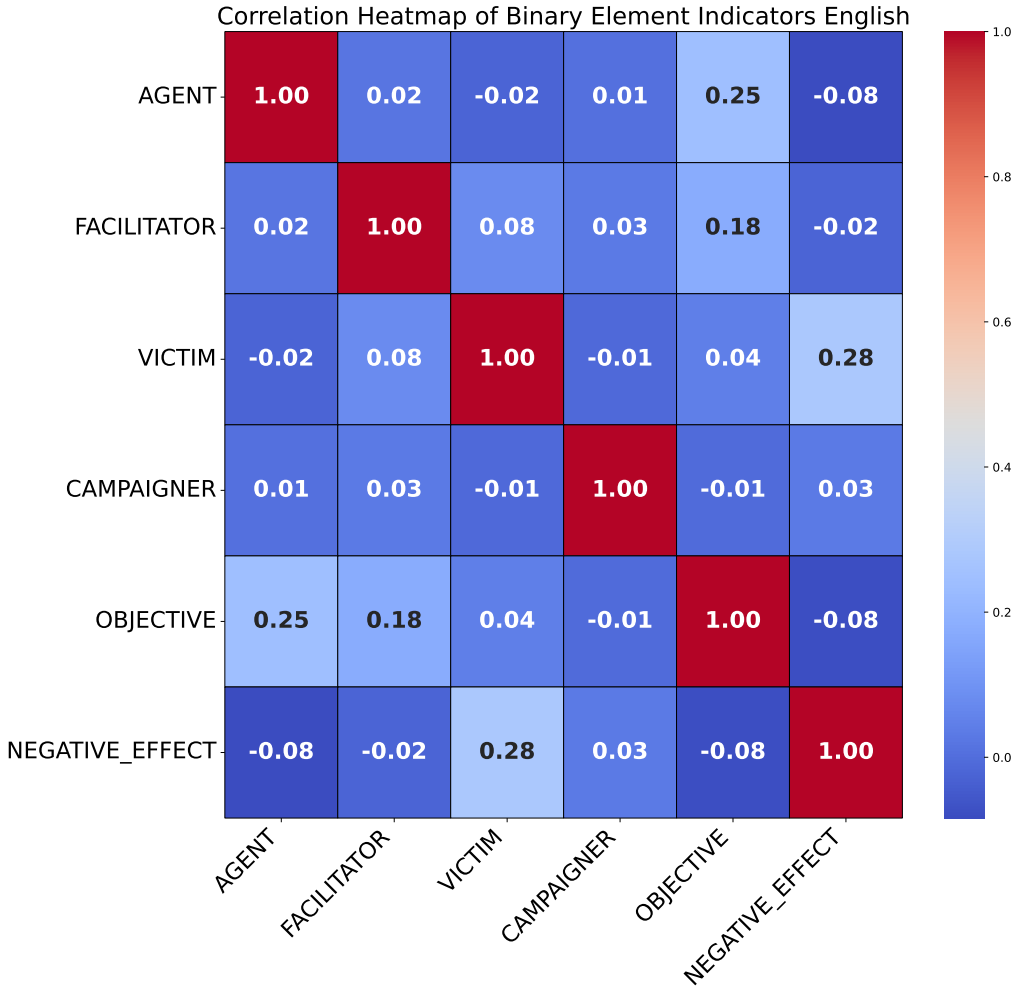


Figure 4.7: Correlation between the presence of at least one span element and other span elements in the English dataset

5 | Evaluation Metrics

This chapter describes the techniques employed to evaluate the models after training for both tasks. Section 5.1 provides a brief overview of the F1-score, a widely used metric in binary classification tasks. Section 5.2 introduces the Matthews Correlation Coefficient (MCC), highlighting its importance in evaluating binary classification models. Finally, Section 5.3 elaborates on Span-F1, a novel metric used for the token classification task.

5.1. Accuracy and F1 Metric

In the field of Natural Language Processing, evaluation metrics such as Accuracy and F1-score are crucial, particularly for tasks involving binary classification.

Accuracy is one of the most straightforward metrics and measures the proportion of correct predictions made by the model. It is defined as the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Here, TP (True Positive) occurs when the model correctly predicts the positive class, TN (True Negative) occurs when the model correctly predicts the negative class, FP (False Positive) happens when the model incorrectly predicts the positive class, and FN (False Negative) occurs when the model incorrectly predicts the negative class.

While accuracy is intuitive and easy to compute, it can be misleading in cases where the dataset is imbalanced. For example, if 95% of the instances belong to one class, a model that always predicts that class will achieve 95% accuracy, despite not effectively distinguishing between classes.

The F1-score is defined as the harmonic mean of Precision (P) and Recall (R), providing a single measure that balances the trade-off between the precision and recall of a model.

Precision measures the accuracy of positive predictions made by the model. It is defined as the ratio of true positive predictions (TP) to the total positive predictions ($TP + FP$).

Mathematically, precision is expressed as:

$$P = \frac{TP}{TP + FP}$$

Recall, on the other hand, measures the model's ability to identify all relevant instances within a dataset. It is defined as the ratio of true positive predictions (TP) to the total actual positive instances ($TP + FN$). The formula for recall is:

$$R = \frac{TP}{TP + FN}$$

Figure 5.1 shows graphically TP, FP, FN, and TN which are used for the calculations.

The F1-score is particularly important in NLP for several reasons. First, it provides a balanced evaluation metric that considers both precision and recall. This balance is essential in NLP tasks where it is critical to maintain high precision without sacrificing recall, and vice versa [56].

Second, in many NLP applications, datasets can be imbalanced, with some classes being underrepresented. The F1-score is more informative than accuracy in such scenarios because it takes into account the model's performance on both the minority and majority classes [41].

Third, the F1-score is the harmonic mean of precision and recall, which gives more weight to lower values. This means that the F1-score will only be high if both precision and recall are high, thus preventing a misleadingly high score in cases where one metric is significantly lower than the other [81].

The F1-score is calculated as follows:

$$F1 = 2 \times \frac{P \times R}{P + R}$$

This formula ensures that the F1-score will be high only when both precision and recall are high. If either precision or recall is low, the F1-score will also be low, reflecting the poor performance of the model in terms of balancing false positives and false negatives.

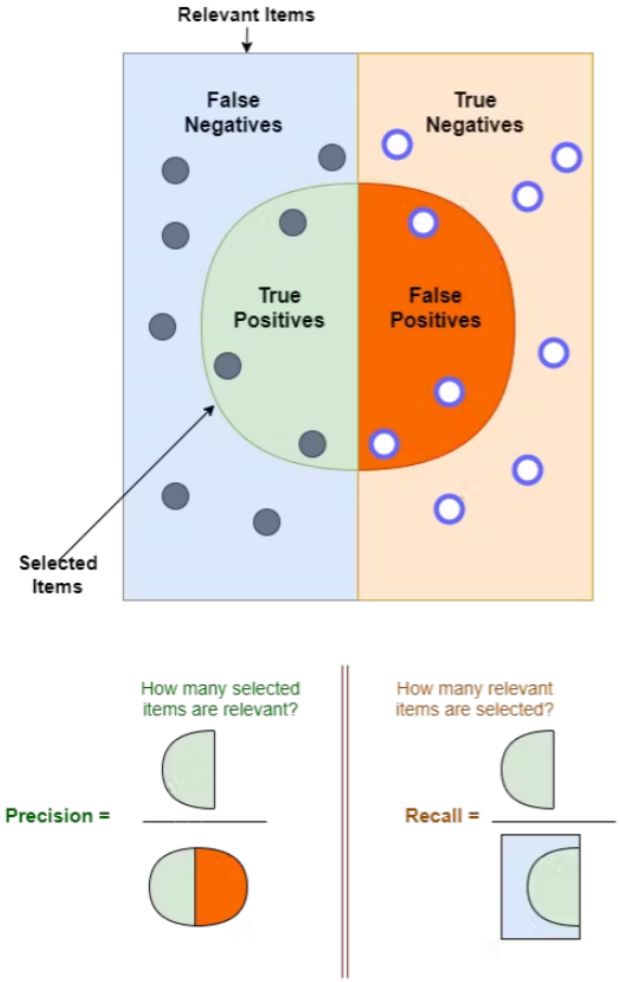


Figure 5.1: True and false positives and negatives used for the evaluation metrics [31]

5.2. Matthews Correlation Coefficient

The Matthews Correlation Coefficient (MCC) [21] is a robust metric used in binary classification tasks to evaluate the quality of predictions. It has become increasingly important in the field of Natural Language Processing due to its balanced consideration of all elements in the confusion matrix: true positives TP, true negatives TN, false positives FP and false negatives FN. MCC is mathematically defined as:

$$\text{MCC} = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

This formula ensures that MCC takes into account the proportion of all four quadrants of the confusion matrix, making it a comprehensive measure. The score ranges from -1 to 1, where 1 indicates a perfect prediction, 0 suggests no better than random chance, and

-1 implies total disagreement between prediction and observation.

In NLP, classification tasks such as sentiment analysis, spam detection, and named entity recognition often deal with imbalanced datasets. Traditional metrics like accuracy can be misleading because they may overestimate the performance of models on datasets with a high prevalence of one class [19, 41].

MCC addresses this imbalance by considering all four confusion matrix components. Unlike the F1 score, which focuses primarily on precision and recall, MCC provides a more holistic evaluation by incorporating true negatives into the assessment. This comprehensive nature makes MCC particularly valuable in NLP applications where the cost of false positives and false negatives is high, and a balanced perspective on classification performance is critical [21, 81].

One of the key advantages of MCC over metrics like accuracy, precision, recall, and even the F1 score is its ability to handle class imbalance more effectively. According to [21], MCC should be preferred in binary classification tasks, especially when the dataset is imbalanced, as it provides a more rigorous and reliable measure of a model's performance. MCC's balanced approach in accounting for all aspects of the confusion matrix makes it a superior choice for evaluating classifiers.

5.3. Span-F1 Metric

Span-F1 is a metric used for the token classification task. It was introduced in [24] for the task on detection of propaganda techniques in news articles at the SemEval evaluation forum in 2020. Span-F1 is particularly important for tasks that involve identifying and classifying spans of text, such as the second task of the PAN challenge on detecting elements of oppositional narratives.

The metric is calculated as follows: Assume d is a text message from a dataset D. Gold spans are contiguous subsequences, t, in a text message: $t \subseteq d$. Each gold span fragment is indicated by a starting and ending integer, corresponding to its location in the text. For example, assuming a text message "Jack Robert Johnson orchestrated a secret plan to manipulate the global stock market.", the corresponding gold span element (in characters) for the Agent ("Jack Robert Johnson") label would be $t = [0, 19]$. Let $T_d = \{t_1, \ldots, t_n\}$ represent all the predefined gold labels t in a text message d while $T = \{T_d\}_{d \in D}$ are the gold labels in the dataset D. Similarly, let $S_d = \{s_1, \ldots, s_n\}$ be all the predicted gold labels in d, and S the same for D. Building on the formulas from [80], [24] define precision (P) and recall (R) as follows:

$$P(S,T) = \frac{1}{|S|} \cdot \sum_{d \in D} \sum_{s \in S_d, t \in T_d} \frac{|s \cap t|}{|t|}$$

$$R(S,T) = \frac{1}{|T|} \cdot \sum_{d \in D} \sum_{s \in S_d, t \in T_d} \frac{|s \cap t|}{|t|}$$

Additionally, the authors define the equations to be equal to zero for $|S| = 0$ and $|T| = 0$ respectively. During prediction, models may generate overlapping predictions for gold span labels. For example, a model may predict that "Jack" ($t = [0, 4]$) is an Agent gold label and also that "Robert Johnson" ($t = [5, 19]$) is as well. Another example could be predicting "Johnson" ($t = [12, 19]$) and "Jack Robert Johnson" ($t = [0, 19]$) as Agents. Since both scenarios are examples of correct predictions, the authors combine adjacent and overlapping solutions before calculating the score. Finally, the F1 score is defined as the harmonic mean between the recall and precision:

$$F1(S,T) = 2 \cdot \frac{P(S,T) \cdot R(S,T)}{P(S,T) + R(S,T)}$$

Span-F1 is crucial for evaluating the performance of models in tasks that require precise identification of text spans. It ensures that both precision and recall are balanced, providing a comprehensive measure of a model's ability to accurately and completely identify relevant spans within the text. This is particularly important in applications such as named entity recognition, where both false positives and false negatives can significantly impact the usefulness of the extracted information [106].

6 | Experimental Framework

In this chapter, we outline the experimental framework developed to train, evaluate, and optimize our models across the two tasks. Section 6.1 begins with a discussion on the rationale behind using cross-validation, specifically explaining the use of Stratified k-fold cross validation to maintain balanced class distributions within each fold. We then delve into the experimental setup in Section 6.2, highlighting the methodological choices made to ensure consistency, reliability, and computational efficiency throughout our process. Subsequently, Section 6.3 will give some insights about hyperparameter optimization.

6.1. Stratified K-Fold Cross Validation

In machine learning tasks, it is common practice to use two different datasets for model development: a training set and a test set. The training set is used to train the model, allowing it to learn patterns from the data, while the test set is held out and used solely for evaluation. This separation ensures that the model's performance is not only optimized for the given data but also generalizes well to new, unseen data, thus preventing overfitting and providing a more accurate measure of its effectiveness.

In the context of the PAN shared task, only the training dataset was available for the majority of the work done in this thesis. Consequently, we had to rely on the training set to evaluate our model. One effective way to create such a dataset is through cross-validation. Specifically, we employed stratified k-fold cross validation, a robust method that offers several advantages for model evaluation and training.

Stratified k-fold cross validation involves partitioning the training set into k subsets, or "folds", of approximately equal size, while maintaining the ratio of class labels within each fold. This stratification ensures that each fold is representative of the overall distribution of the data, which is particularly important in imbalanced datasets where some classes may be underrepresented. During each epoch of training, the model is trained k times, where $k - 1$ folds are used for training and the remaining fold is used for validation. The model's performance is then assessed by averaging the results across all k folds and

epochs, providing a comprehensive evaluation metric.

An essential part of this cross-validation process is the use of model checkpoints, where the best-performing model in each fold (based on validation metrics) is saved. This approach allows for careful monitoring of the model's performance during training and ensures that the best version of the model within each fold is preserved for final evaluation. By saving checkpoints, we could retrieve models trained under optimal conditions, thus avoiding potential overfitting from training on all folds without validation.

After training on each fold, an alternative way to calculate the final performance metric for the model is obtained by averaging the results from all the best checkpoints across the k folds. This averaged score offers a reliable and stable measure of model performance, reflecting the model's generalizability across different subsets of the training data.

The implementation of stratified k-fold cross validation used in this thesis was facilitated by the Scikit-learn library [73]. For both binary and token classifiers, we used three folds ($k = 3$). This choice strikes a balance between computational efficiency and the robustness of the validation process.

Cross-validation was crucial for obtaining reliable performance scores for our models in the absence of the official test dataset.

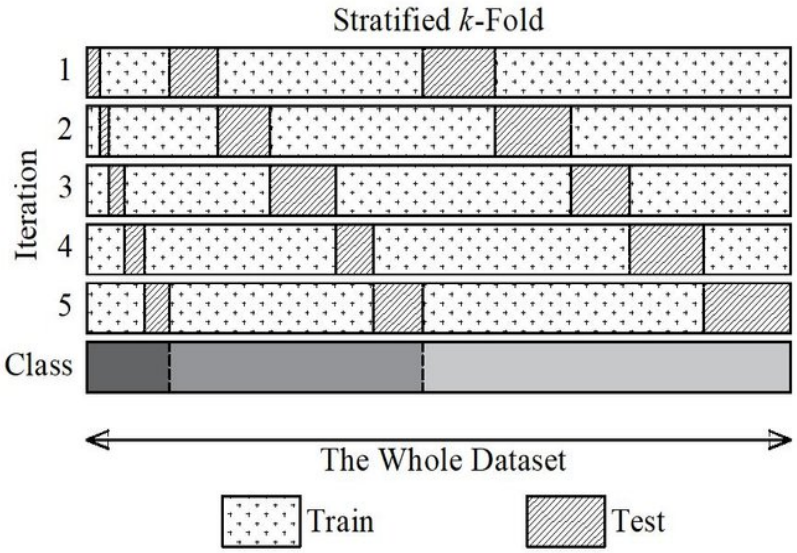


Figure 6.1: Stratified k-fold cross validation. Example with k=5 [5]

6.2. Experiments Setup

Experiments were conducted on three NVIDIA GeForce GTX 1080 (8 GB memory). To ensure uniformity and comparability of results, the same experimental setting was consistently applied across all tasks, runs, and languages under study. Our experimental setup, designed for optimal performance and efficient resource utilization, utilized stratified k-fold cross validation with 3 folds, as detailed in Section 6.1, to ensure robust performance across different subsets of the data.

The training process spanned 15 epochs. A weight decay of 0.01 was applied as a regularization technique to penalize large weights. A custom linear learning rate scheduler was employed, adjusting the learning rate from an initial value of 2e-5 to a final value of 2e-6 over the total number of training epochs. Gradient accumulation steps were set to 4, effectively increasing the batch size without inflating the memory footprint by accumulating gradients over multiple steps before updating the model's weights. The training batch size per device was dynamically set based on available GPU memory, managed by a custom callback designed to dynamically adjust the batch size used during training and evaluation based on the available GPU memory. This ensures efficient resource utilization and prevents memory-related issues during training, especially when dealing with varying data sizes and model complexities. This adaptive approach ensured optimal resource utilization without encountering memory constraints, thereby maintaining high computational efficiency and preventing potential memory-related issues during training. Upon completion of training, the best model, as determined by the F1 score, was loaded.

In the following table we can better observe the experiment setting:

learning rate	linear_learning_rate_scheduler
evaluation_strategy	epoch
save_strategy	epoch
num_train_epochs	15
weight_decay	0.01
load_best_model_at_end	True
metric_for_best_model	f1
gradient_accumulation_steps	4
per_device_train_batch_size	batch_size
per_device_eval_batch_size	batch_size

Table 6.1: Experiments setup for all the tasks

6.3. Hyperparameter Optimization

To enhance the performance of our binary classification models, we integrated two prominent hyperparameter optimization techniques: Grid Search and Bayesian Optimization [2]. Initial experiments were conducted using more compact transformer architectures, such as distilbert-base-cased[1] for English and bert-base-spanish-wwm-uncased[2] [18] for Spanish, to assess the effects of these optimization approaches. However, the outcome did not meet expectations. The increase in F1-score was minimal (around 0.1%), while the training duration grew significantly. This finding is consistent with existing studies, which often report that hyperparameter tuning for transformers rarely produces substantial gains for datasets containing only a few thousand samples.

The prevailing consensus in the community acknowledges that default hyperparameter configurations for models like BERT generally provide satisfactory performance for many applications. The computational cost and training time associated with exhaustive hyperparameter optimization are typically disproportionate to the incremental improvements achieved. Our results reinforced this understanding, highlighting the considerable resource demands without a corresponding payoff in performance metrics [2].

Consequently, based on our observations and a practical cost-benefit analysis, we decided against extensive hyperparameter tuning in further experiments. Instead, we prioritized exploring more impactful strategies that could yield significant improvements without imposing heavy computational burdens.

[1] https://huggingface.co/distilbert/distilbert-base-cased
[2] https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased

7 | Binary Classification of Conspiratorial vs. Critical Thinking

In this chapter, we will provide a detailed account of the experiments conducted for the binary classification task, beginning with the data augmentation techniques applied to the dataset. This will be followed by a comprehensive analysis of the experiments carried out on the train dataset, culminating in a discussion of two systems submitted for the competition.

7.1. Data Augmentation

Data augmentation was crucial in enhancing the effectiveness of the binary classification models, especially when facing with challenge like imbalanced datasets, as in our case. In such scenarios, where one class often dominates, models are prone to overfitting and biased predictions. Data augmentation techniques like back-translation, summarization, or paraphrasing were employed to generate varied text while preserving the original meaning. Artificially expanding the dataset enhance the model's exposure to different linguistic structures and expressions, allowing it to better capture subtle patterns in both classes, particularly the minority class. By enriching the dataset in this way, data augmentation not only improves classification performance but also helps create more resilient models that can handle real-world linguistic variability, leading to more accurate and reliable predictions.

7.1.1. Summarization

A first experiment involved summarizing the datasets using different models, such as T5-small[1], T5-base[2], and T5-large[3] [70]. Summarizing with T5-small yielded sub-optimal results due to its limited capacity, which restricted its ability to capture and compress the full context of longer or more complex texts. It struggled to balance abstraction and comprehension, resulting in outputs that merely repeated or extracted a subpart of the original text without effectively rephrasing or condensing it. The model lacked the depth to perform true abstractive summarization, where new, shortened representations are created while maintaining the full text's meaning and key details, leading to less informative and often incomplete summaries. While T5-base showed some improvement, generating slightly more balanced summaries, and T5-large performed even better with more comprehensive and coherent outputs, the results were still unsatisfactory. They fell short of ideal performance, particularly with complex, nuanced texts, where the generated summaries still missed key details or lacked the refinement expected in human-level summarization. A recurring issue across all these models, especially T5-small, was the frequent lack of coherence, producing sentences that were awkward or nonsensical. Despite the larger models having more parameters to work with, they also struggled to maintain the logical flow of information, resulting in summaries that missed key details and sometimes made little sense, ultimately failing to meet expectations for high-quality summarization.

7.1.2. Paraphrasing

Since summarization yielded poor results, we turned to paraphrasing using the fine-tuned T5-large model. The T5 model automatically handles all the steps, including text cleaning and back-translation techniques. It processes the input text by removing unwanted characters and spaces, then introduces variations through its built-in back-translation mechanisms. The model translates the text to a secondary language (internally managed) and back, ensuring the paraphrased output remains close to the original meaning while diversifying sentence structures.

However, the results were not good enough to incorporate the newly paraphrased dataset into further experiments. The paraphrased text frequently suffered from disrupted word order, leading to nonsensical or grammatically incorrect sentences. Rather than producing meaningful variations, the model often misplaced key words or phrases, which interfered with the logical flow of the sentences and made them difficult to interpret. This issue

[1] https://huggingface.co/google-t5/t5-small
[2] https://huggingface.co/google-t5/t5-base
[3] https://huggingface.co/google-t5/t5-large

likely stemmed from problems in the tokenization and decoding process of the T5 model, where sentences were split into subword tokens that the model struggled to recombine coherently, resulting in broken sentence structures.

Moreover, the built-in back-translation process may have introduced subtle shifts in sentence construction that further confused the paraphrasing model. These shifts, while introducing variation, also altered the original sentence structure in ways that made it harder for the model to reconstruct a coherent paraphrase.

7.1.3. One-Way Translation

After the initial data augmentation techniques yielded unsatisfactory results, we turned to one-way translation as the final approach, taking advantage of the fact that we had two distinct datasets, one in English and one in Spanish. This process involved translating each dataset into the other language, effectively doubling the amount of data available for training. In the first trial, we employed the *googletrans* Python library[4], which serves as an API wrapper for Google Translate. While this approach produced satisfactory results, further testing with the Helsinki-NLP models yielded even better performance.

For the translations, we used the MarianMT[5] model to automate the process. The *Helsinki-NLP/opus-mt-es-en*[6] model was employed to translate the Spanish dataset into English, while the *Helsinki-NLP/opus-mt-en-es*[7] model was used for translating the English dataset into Spanish [95]. The translation process began by loading the datasets, and for each entry, both the main text and any associated annotations were translated. The MarianMT model tokenized the input text, generated the translation, and decoded the output into the target language.

Translating the datasets allowed the creation of two expanded versions of the original datasets, thus effectively doubling the size of the training data. This augmentation technique significantly enriched the dataset by introducing more linguistic diversity and a wider range of sentence structures. By exposing the models to both the original and translated versions of each dataset, the models were able to learn from a greater variety of linguistic patterns, improving their ability to generalize to unseen data. Figures 7.1 and 7.2 provide examples of entries translated from Spanish to English and from English to Spanish, respectively, demonstrating how the core meaning of the text was preserved.

[4] https://pypi.org/project/googletrans/
[5] https://huggingface.co/docs/transformers/model_doc/marian
[6] https://huggingface.co/Helsinki-NLP/opus-mt-es-en
[7] https://huggingface.co/Helsinki-NLP/opus-mt-en-es

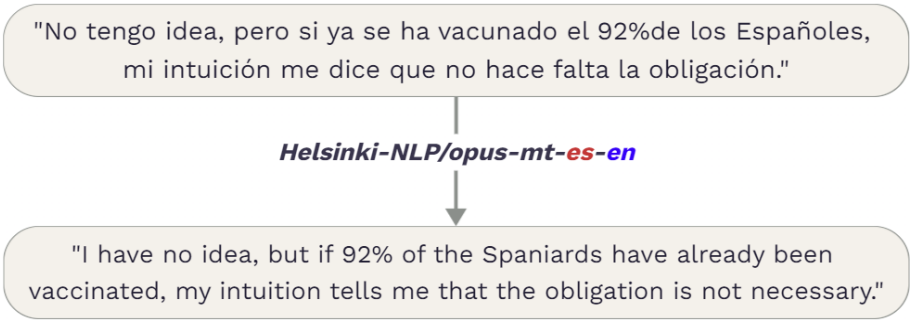


Figure 7.1: Example of translation from Spanish to English

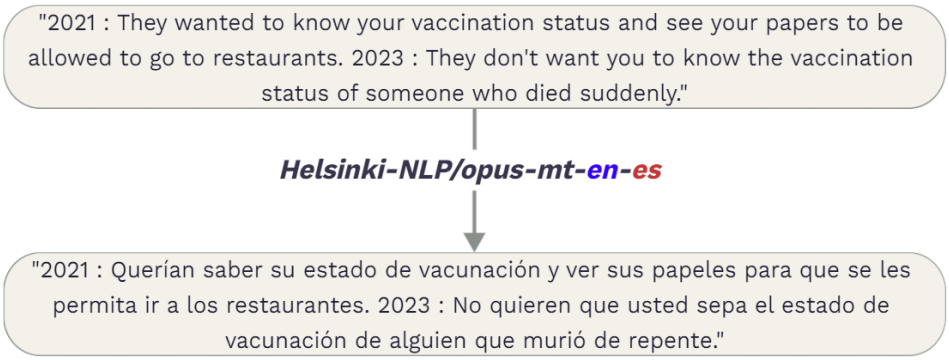


Figure 7.2: Example of translation from English to Spanish

This strategy was particularly valuable in this context because the Spanish and English datasets were not identical in content, meaning that translating each dataset into the other language provided additional, complementary data rather than just duplicating existing examples. By translating both datasets, the models benefited from more varied training data, which helped to balance the representation of both languages in the model's learning process.

7.2. Data Processing

Data preprocessing and postprocessing were important to prepare raw data for modeling in the case of data preprocessing and refine the model's output, ensuring that the predicted class labels are practical and interpretable, in case of data postprocessing. Together, these processes ensure that binary classification models are trained on high-quality data and produce outputs that are accurate, actionable, and aligned with specific application requirements. The specific techniques and their importance for binary classification will be detailed further in Section 7.2.1 for preprocessing and Section 7.2.2 for postprocessing.

7.2.1. Data Preprocessing

The data preprocessing began with loading the dataset from a JSON file. Next, the labels "CONSPIRACY" and "CRITICAL" were mapped to numerical values (0 and 1). This allows the model to treat the labels as binary classification targets. Then, the model tokenizer broke the text into tokens, applying padding and truncation to ensure uniform input lengths for efficient batch processing.

7.2.2. Data Postprocessing

Since we tackled two distinct NLP tasks, each required specific postprocessing steps to ensure that the model outputs conformed to the format required by the organizers. For the binary classification task, a series of detailed procedures were applied to refine the model predictions and store them effectively. The two submitted systems will be explained in Section 7.5.

In the first submission, referred to as the conservative approach, we utilized a custom fine-tuned model for sequence classification, along with its associated tokenizer. This step ensured that the model could accurately interpret and classify the input texts. Following the loading of the model and tokenizer, a text classification pipeline was established. This pipeline facilitated the application of the model to the test dataset by managing the tokenization of input texts, processing these texts through the model, and collecting classification results. The pipeline's automation of these steps ensured efficient processing and consistent handling of the test data.

For the second submission, the experimental approach, we employed three fine-tuned models and their respective tokenizers, each tasked with predicting the category of a given text. After obtaining predictions from all three models, the final category for each document was determined by selecting the label with the highest averaged probability. Finally, these numerical predictions were mapped back to the corresponding labels, either conspiracy or critical.

7.3. Systems

In this section we will describe all the different systems applied in our approach. First would be describe the transformers based approach (Subsection 7.3.1), then we will explain two different classifiers with different text preprocessing techniques (Subsection 7.3.2), and them we will show all the combination tried in the ensembling approach (Subsection 7.3.3).

7.3.1. Transformer-based Approach

The first approach was fine-tuning different transformers models. The process started by ensuring the reproducibility of results, setting a fixed random seed across different components, ensuring that all experimental runs were consistent and produced the same results under identical conditions.

The training procedure was organized using a 3-fold stratified cross-validation setup. Important aspects were the system's ability to dynamically adjust the batch size based on available GPU, the custom learning rate scheduler that gradually reduces the learning rate throughout the training process, as described in details in Section 6.2. The training process was organized over multiple epochs, and at the end of each epoch, the model's performance was evaluated using a variety of metrics, including accuracy, precision, recall, and F1 score. The results from each fold were written to files, allowing for detailed analysis and comparison across different models or experimental setups. By saving the best-performing model from each fold, the system ensured that the best version of the model is preserved for future use or deployment.

7.3.2. Classifiers

We also tried two different classifiers, each coupled with different text preprocessing techniques. The first classifier, referred to as *clf1*, operates using a CountVectorizer to transform raw textual data into a matrix of word counts, where each document is represented by the frequency of its constituent terms. This matrix is then used to train a Multinomial Naive Bayes model, a probabilistic classifier that assumes conditional independence between features. The second classifier, *clf2*, implements a more sophisticated vectorization technique, TF-IDF Vectorizer, which assigns weights to terms based on their frequency in individual documents relative to their appearance across the entire dataset. This approach diminishes the impact of commonly occurring words that offer limited semantic value, while emphasizing rarer, more contextually important terms. Logistic Regression, a discriminative model, is then applied to these weighted features, estimating the likelihood of a document belonging to a particular class based on the learned relationships between words and categories.

Accompanying these classifiers are three different text preprocessing pipelines, each with varying levels of complexity. The simplest approach, referred to as *basic*, converts the text to lowercase, ensuring uniformity across the dataset while retaining punctuation, stop words, and numbers. This method, though computationally light, preserves much of the noise inherent in raw text. The second approach, denoted as *spacy*, leverages the

capabilities of the spaCy library to perform advanced linguistic preprocessing [4]. This includes tokenization, lemmatization, and the removal of stop words and punctuation, allowing the model to focus on the core semantic content of each document by reducing words to their base forms and filtering out irrelevant tokens. In contrast, the third and most complex preprocessing strategy, known as *spacy_pos*, further refines the input data by applying part-of-speech tagging to retain only tokens belonging to specific grammatical categories, such as nouns, verbs, adjectives, adverbs, and proper nouns. This selective filtering enhances the model's ability to concentrate on syntactically meaningful components of the text.

In order to use classifiers as fine-tuned models within an ensemble, explained in the next subsection, wrapping the classifier was essential for making it reusable and adaptable for different tasks. The wrapping approach used encapsulated the entire process of text preprocessing, feature extraction, and model training into a single, self-contained object. This design allowed the classifier to be treated like any other fine-tuned model. Once wrapped, the classifier could easily be loaded and reused in various parts of the project without having to repeat the setup process. For instance, when used in an ensemble, the classifier takes processed input data and generates predictions that are combined with predictions from other models. The wrapping ensures that all aspects of the classifier's behavior, from text transformation to prediction generation, are handled within a single interface.

7.3.3. Ensemble Models

The final approach explored was an ensemble method, where we tried numerous combination of fine-tuned models and classifiers, using both Soft and Hard voting strategies. After loading the models and their respective tokenizers, the dataset was processed in small batches to optimize memory usage. For each batch, the input texts were passed through all three models individually, generating predictions. In soft voting, the prediction for each text consisted of the probability distribution over the two classes. The final prediction for each text was determined by averaging these probabilities across the models, and the class with the highest average probability was selected as the final prediction. In contrast, hard voting operated directly on the predicted categories, instead of averaging probabilities, each model made a categorical prediction. The final prediction for each text was then determined by majority vote. The class predicted by most of the models was chosen. Once the predictions were aggregated, accuracy, precision, recall and F1-score were calculated. They provided a comprehensive assessment of the model's performance, highlighting the strengths of both voting strategies depending on the task.

7.4. Experiments on the Training Dataset

In Table 7.1, we present the results for the English dataset. The second column specifies the number of epochs each model was trained for, while "DA" in the third column stands for "Data Augmented." We represent the first classifier with *clf1* and the second one with *clf2*, each employing different preprocessing techniques: basic, spaCy, or spaCy with part-of-speech tagging (spacy_pos). Only ensembles involving the second classifiers are reported in the table, as they yielded better results. In the *ensemble-2_models-1_clf2*, the two transformer models used were *roberta-base (15 epochs)* and *bert-base-uncased (15 epochs)*, combined with the *clf2-spacy_pos (DA)* classifier. In the *ensemble-1_model-2_clf2*, the transformer model was *roberta-base (15 epochs)*, and the two classifiers involved were *clf2-spacy (DA)* and *clf2-spacy_pos (DA)*. The abbreviations *HV* and *SV* denote hard voting and soft voting ensemble methods, respectively.

The data shows that transformer-based models, especially ensemble models, generally perform better across key metrics. For instance, the *ensemble-3-models_SV* achieves the highest accuracy (0.9041), best F1 score (0.9134), and highest precision (0.9223). The *ensemble-3-models_HV* also performs strongly with an accuracy of 0.9031 and an F1 score of 0.9101. Among the individual transformer models, *roberta-base (15 epochs)* stands out with strong performance, achieving an accuracy of 0.8793 and an F1 score of 0.9072.

Traditional classifiers, such as the *clf2-basic (DA)* model, deliver solid results as well, with an accuracy of 0.884 and an F1 score of 0.886. The performance of *clf2-spacy (DA)* is even better, reaching 0.888 in accuracy and 0.890 in F1 score. These numbers indicate that while traditional classifiers do not outperform the transformer models, they remain competitive, especially when enhanced with spaCy-based preprocessing and data augmentation. Despite their simpler architectures, these classifiers can yield respectable results. The *clf2-spacy (DA)* model, for example, nearly matches some transformer-based models in both accuracy and F1 score. This suggests that with the right enhancements, such as data augmentation and feature engineering, traditional classifiers remain a relevant choice. Their lower computational requirements make them particularly useful when resources are limited or when transformer models are impractical.

Moreover, ensembling the models leads to better results for both transformers and classifiers. This demonstrates that combining the strengths of multiple models, regardless of the underlying architecture, results in stronger overall performance.

Model	Epochs	DA	Acc.	F1	F1-macro	Prec.	Recall
scibert_scivocab_uncased	15	✓	0.8637	0.8973	0.8475	0.8734	**0.9225**
bert-base-uncased	15	✓	0.8688	0.9005	0.8539	0.8813	0.9205
roberta-base	15	✓	0.8793	0.9072	0.8672	0.8994	0.9151
bert-large-uncased	5	✓	0.8631	0.8638	0.8401	0.8645	0.8631
distilbert-base-uncased	5	✓	0.8629	0.8637	0.8518	0.8640	0.8633
distilbert-base-uncased	3	✓	0.7901	0.7891	0.7730	0.7887	0.7895
ensemble-3-models_HV	-	✓	0.9031	0.9101	0.8801	0.9181	0.9022
ensemble-3-models_SV	-	✓	**0.9041**	**0.9134**	**0.8916**	**0.9223**	0.9048
CT-bert	4	✓	0.7701	0.7412	0.7102	0.7507	0.7317
clf1-basic	-		0.846	0.881	0.831	0.875	0.887
clf1-spacy	-		0.849	0.846	0.831	0.845	0.847
clf1-spacy_pos	-		0.844	0.849	0.824	0.842	0.856
clf2-basic	-		0.848	0.844	0.821	0.856	0.844
clf2-spacy	-		0.865	0.848	0.844	0.863	0.854
clf2-spacy_pos	-		0.866	0.868	0.843	0.872	0.860
clf1-basic	-	✓	0.849	0.851	0.834	0.853	0.848
clf1-spacy	-	✓	0.860	0.862	0.843	0.867	0.854
clf1-spacy_pos	-	✓	0.858	0.861	0.841	0.868	0.852
clf2-basic	-	✓	0.884	0.886	0.870	0.885	0.889
clf2-spacy	-	✓	0.888	0.890	0.872	0.888	0.893
clf2-spacy_pos	-	✓	0.886	0.887	0.870	0.888	0.892
ensemble-3-clf2_HV	-	✓	0.720	0.725	0.612	0.743	0.724
ensemble-3-clf2_SV	-	✓	0.731	0.718	0.644	0.738	0.724
ensemble-2_models-1_clf2_HV	-	✓	0.889	0.871	0.865	0.879	0.869
ensemble-2_models-1_clf2_SV	-	✓	0.885	0.867	0.861	0.870	0.865
ensemble-1_model-2_clf2_HV	-	✓	0.887	0.865	0.859	0.871	0.856
ensemble-1_models-2_clf2_SV	-	✓	0.838	0.881	0.874	0.887	0.874

Table 7.1: Binary classification models performance metrics for the English train dataset

Table 7.2 shows the results for the Spanish dataset. As before, only ensembles involving the second classifiers are included in the table. In the *ensemble-2_models-1_clf2*, the two transformer models used were *bert-base-spanish-wwm (15 epochs)* and *bertin-roberta-base-spanish (15 epochs)*, combined with the *clf2-spacy_pos (DA)* classifier. In the *ensemble-1_model-2_clf2*, the transformer model was *bert-base-spanish-wwm (15 epochs)*, and the two classifiers involved were *clf2-spacy (DA)* and *clf2-spacy_pos (DA)*.

The table highlights that transformer-based models, especially ensemble models, consistently outperform traditional classifiers. The *ensemble-3_models_SV* achieves the highest accuracy (0.8971), best F1 score (0.9248), and highest precision (0.9020). The *ensemble-3_models_HV* also performs strongly with an accuracy of 0.8802 and an F1

score of 0.9172. These results demonstrate the clear benefit of ensembling. Individual transformer models like *bert-base-spanish-wwm (15 epochs)* and *bsc-bio-ehr-es-pharmaconer (15 epochs)* deliver solid performance but are still outshined by the ensemble approaches.

Traditional classifiers, although generally behind the transformers, perform competitively when enhanced with data augmentation. For example, the *classifier2-spacy (DA)* model achieves an accuracy of 0.896 and an F1 score of 0.885, showing that older methods can still be effective when improved with modern techniques. This suggests that traditional classifiers remain a viable option, especially when computational resources are limited.

Model	Epochs	DA	Acc.	F1	F1-macro	Prec.	Recall
bertin-roberta-base-spanish	15	✓	0.8613	0.8992	0.8384	0.8462	**0.9593**
bert-base-spanish-wwm	15	✓	0.8687	0.9005	0.8539	0.8813	0.9205
bsc-bio-ehr-es-pharmaconer	15	✓	0.8669	0.8973	0.8542	0.8934	0.9012
roberta-base-bne	10	✓	0.8675	0.8674	0.8550	0.8672	0.8675
roberta-base	5	✓	0.8555	0.8529	0.8369	0.8546	0.8555
ensemble-3_models_SV	-	✓	**0.8971**	**0.9248**	**0.8841**	**0.9020**	0.9488
ensemble-3_models_HV	-	✓	0.8802	0.9172	0.8798	0.8910	0.9451
alberto-base-spanish	-	✓	0.8612	0.8610	0.8428	0.8609	0.8613
classifier1(basic)	-	-	0.807	0.7945	0.784	0.792	0.797
classifier1-(spacy)	-	-	0.814	0.7990	0.788	0.800	0.798
classifier1-(spacy_pos)	-	-	0.816	0.7994	0.789	0.804	0.795
classifier2-(basic)	-	-	0.809	0.7886	0.774	0.805	0.773
classifier2-(spacy)	-	-	0.818	0.7985	0.772	0.818	0.780
classifier2-(spacy_pos)	-	-	0.829	0.8118	0.783	0.836	0.789
classifier1-(basic)	-	✓	0.871	0.8604	0.810	0.857	0.864
classifier1-(spacy)	-	✓	0.870	0.8575	0.812	0.858	0.857
classifier1-(spacy_pos)	-	✓	0.875	0.8625	0.815	0.864	0.861
classifier2-(basic)	-	✓	0.877	0.886	0.861	0.891	0.881
classifier2-(spacy)	-	✓	0.896	0.885	0.864	0.896	0.875
classifier2-(spacy_pos)	-	✓	0.894	0.883	0.862	0.893	0.874
ensemble-3-clf2_HV	-	✓	0.772	0.748	0.724	0.788	0.712
ensemble-3-clf2_SV	-	✓	0.798	0.778	0.729	0.815	0.745
ensemble-2_models_1_clf_HV	-	✓	0.818	0.799	0.734	0.831	0.771
ensemble-2_models_1_clf_SV	-	✓	0.832	0.818	0.783	0.852	0.786
ensemble-1_model_2_clf_HV	-	✓	0.820	0.803	0.749	0.836	0.773
ensemble-1_models_2_clf_SV	-	✓	0.838	0.823	0.764	0.848	0.799

Table 7.2: Binary classification models performance metrics for the Spanish train dataset

7.5. Submitted Systems for Task 1

This section outlines the systems submitted for task 1, which focuses on distinguishing between critical and conspiracy texts. For this task, the general approach involved fine-tuning transformer-based models and applying good data processing, described in Section 7.2 and data augmentation techniques, described in Section 7.1. Both English and Spanish datasets were processed similarly. The main difference between Run 1 and Run 2 lies in the method used to make predictions. In Run 1, only the best model checkpoint was used to make predictions while in Run 2, an ensembling method was employed, which combined the predictions from multiple models.

In Run 1, we focused on using the best model checkpoint to make predictions for both English and Spanish datasets. For English, the *facebook/roberta-base*[8] [59] model was used. This model is known for its robust performance on various NLP tasks. For Spanish, the *dccuchile/bert-base-spanish-wwm-uncased* model was used, which is specifically trained for the Spanish language.

In Run 2, an ensembling approach was used. The general approach involved using a Soft Voting Ensembling method composed of three custom fine-tuned transformer-based models. We selected two well-known transformer models and incorporated a scientific one. The inclusion of a scientific model aimed to better capture the specialized terminology and nuanced discussions related to scientific content and misinformation about COVID-19. We used the following three models for English:

1. *facebook/roberta-base*[59]
2. *google/bert-base-uncased*[9] [27]
3. *allenai/sciber_ scivocab_ uncased*[10] [8]

For the Spanish dataset, the approach mirrored the English task but involved different custom fine-tuned models suited to the Spanish language that are:

1. *dccuchile/bert-base-spanish-wwm-uncased*[18]
2. *PlanTL-GOB-ES/bsc-bio-ehr-es-pharmaconer*[11] [17]
3. *bertin-project/bertin-roberta-base-spanish*[12] [26]

[8] https://huggingface.co/FacebookAI/roberta-base
[9] https://huggingface.co/google-bert/bert-base-uncased
[10] https://huggingface.co/allenai/scibert_scivocab_uncased
[11] https://huggingface.co/PlanTL-GOB-ES/bsc-bio-ehr-es-pharmaconer
[12] https://huggingface.co/bertin-project/bertin-roberta-base-spanish

For ensembling, the best checkpoint from each fine-tuned Spanish model was used in a Soft Voting ensemble for predictions. This approach involved combining predictions from multiple models using a Soft Voting ensemble [61], which significantly improved the overall accuracy and robustness of the system. The Soft Voting process involves averaging the predicted probabilities of each category from the different models and then making the final prediction based on the highest average probability.

The methods and findings discussed in this chapter provide insights into **RQ1**, specifically evaluating model performance in distinguishing conspiracy from critical thinking narratives.

8 | Span-level Detection of Narrative Elements

In this chapter, we will provide a detailed account of the experiments conducted for the span-level detection task, beginning with the data augmentation techniques applied to the dataset. This will be followed by the data preprocessing and the comprehensive analysis of the experiments carried out on the train dataset, culminating in a discussion of the models developed and submitted for the competition.

8.1. Data Augmentation

Also in this case, data augmentation was crucial in improving the performance of the token classification models. In token classification, certain token classes often dominate the dataset, leading to overfitting and skewed predictions. To mitigate this issue, we employed several augmentation strategies aimed at diversifying the dataset while preserving the core semantics of the text.

8.1.1. One-way Translation

The initial experiment involved using the translated datasets from the binary classification, with the addition of translating also the "span-text" associated with the six distinct span categories. However, we encountered a significant challenge: after translation, the "start_char" and "end_char" markers, which denote the beginning and end positions of these spans, no longer corresponded to the precise words in the original text, as well as "start_token" and "end_token". This misalignment arose because there isn't a one-to-one match in the length of the characters between the original and translated texts. Consequently, the span annotations became inaccurate, rendering this approach unusable for our purposes.

8.1.2. Synonym Replacement

The second approach was using synonym replacement. This technique is very efficient in token classification because it maintains the structure of the input while introducing vocabulary diversity. In tasks where tokens are mapped to specific labels, keeping the same number of words ensures that the start and end token positions remain aligned. This consistency is crucial for avoiding misalignment in token spans, which could lead to incorrect classification.

When synonyms are used while preserving the token count, the model benefits from encountering different words with similar meanings without disrupting the original token structure. This increases the model's ability to generalize, making it more robust when dealing with variations in natural language. Additionally, synonym translation enriches the dataset by introducing new vocabulary, offering the model a broader context to understand linguistic nuances, without needing adjustments to the underlying tokenization or the training process.

The first trial of synonym replacement was using WordNet from the NLTK library[1]. WordNet, a comprehensive lexical database, organizes words into *synsets* that group together words with similar meanings. By accessing these synsets, alternative lemmas (word forms) can be extracted, offering a range of possible replacements for a given word. In this method, words in both the main text and associated span annotations are replaced with randomly selected synonyms from WordNet, generating alternative versions of the original sentences while preserving the core meaning. However, the method does not take into account the specific context of each word, leading to inappropriate or awkward replacements when a word has multiple meanings. In addition this approach was highly conservative, resulting in minimal changes to the text. WordNet's synsets provided only a limited number of suitable synonyms, and as a result, most words in both the main text and span annotations remained unchanged. This conservative nature meant that very few words were substituted, leading to a dataset that was nearly identical to the original. Due to the lack of significant variation introduced by this method, the augmented dataset was essentially a copy of the original, rendering it ineffective for use in further experiments. The intended goal of generating diverse linguistic structures to improve model generalization was not achieved. Consequently, the augmented data could not be used in training, as it offered no meaningful distinction from the original dataset, limiting the overall impact of the augmentation strategy.

The next approach was using spaCy's pre-trained word embeddings [4]. It begins by

[1] `https://www.nltk.org/howto/wordnet.html`

loading the *"en_core_web_md"* spaCy model, which includes pre-trained word vectors that allow for the calculation of word similarities. For each word in the dataset's text, a synonym is identified by comparing its vector representation with those of other words in the vocabulary. A synonym is selected only if it has the same part-of-speech tag as the original word, ensuring that the new word fits grammatically within the context. If no suitable synonym is found, the original word remains unchanged. After modifying the text, the approach checks if the total word count remains the same as the original. If the word count differs, which indicates that the replacement disrupted the sentence structure, that particular item is skipped to maintain consistency. However, the synonym replacement process was very conservative, often leaving many words unchanged due to strict part-of-speech and similarity requirements. Additionally, the process of performing the synonym translation was notably slow, as identifying and replacing synonyms for each word required considerable computational time.

The last approach for enhancing the English dataset involved a combination of advanced tools such as spaCy, static word embeddings (word2vec) [65], and Google News word vectors. This strategy began with spaCy parsing the text to segment each word and assigning the appropriate part of speech, creating a structured linguistic base from which to work. Subsequently, the Google News word2vec vectors, which were compiled from a vast array of news articles, provided a rich network of lexical associations. These vectors are specifically the GoogleNews-vectors-negative300 static word embeddings[2], trained on a dataset of 100 billion words from Google News articles. This allowed for the precise identification of synonyms that closely matched the semantic context of the original words, ensuring that replacements were both meaningful and contextually appropriate. The carefully selected synonyms were then integrated into the text, followed by a crucial recalibration of annotations for text spans to ensure the annotations accurately reflected the changes in word positions and meanings, thus preserving the dataset's relevance and usefulness.

An example of the synonym augmented data for the English dataset is illustrated in Figure 8.1.

[2] `https://github.com/mmihaltz/word2vec-GoogleNews-vectors`

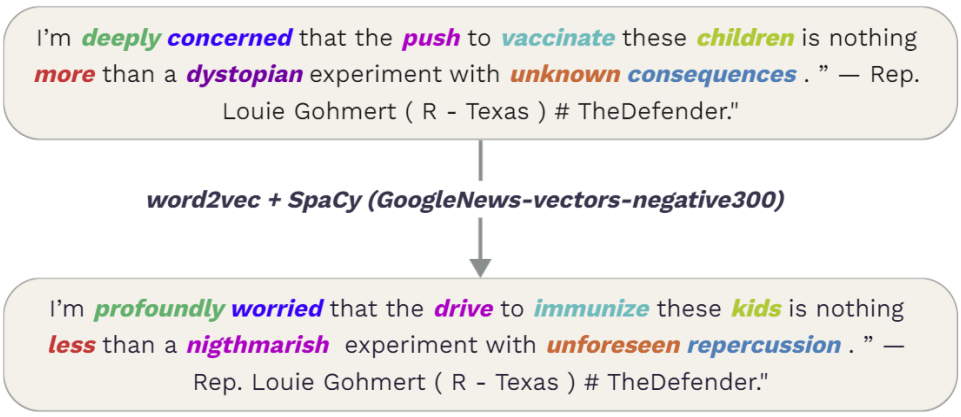


Figure 8.1: Synonym augmented English dataset example

Building upon the method utilized for the English dataset, a similar approach was adopted for the Spanish dataset, specifically tailored to accommodate the distinct linguistic characteristics of Spanish. This process employed the *"es_core_news_sm"* spaCy model for precise linguistic parsing and syntactic tagging, essential for effective synonym replacement. Further enhancing the approach, *FastText* embeddings from the Spanish Unannotated Corpora[3] were integrated. This dataset, in contrast to the Google News vectors, includes a corpus size of 3 billion words and was selected since GoogleNews-vectors were available only for the English dataset. This ensured that the chosen synonyms were contextually appropriate and semantically aligned with the original text.

As with the English process, the Spanish text underwent analysis with the spaCy model to identify grammatical structures, followed by the FastText model's assistance in selecting suitable synonyms for nouns, verbs, adjectives, and adverbs. Special attention was paid to maintain the original intent and readability of the text, carefully avoiding multi-word synonyms that could complicate the narrative flow. Upon integrating the appropriate synonyms, the next critical step involved recalibrating the text's annotations to ensure that they correctly corresponded with the updated text, thus ensuring the annotations remained accurate and relevant for future processing and analysis tasks. An example of the synonym-augmented data for the Spanish dataset is shown in Figure 8.2.

[3] `https://github.com/dccuchile/spanish-word-embeddings`

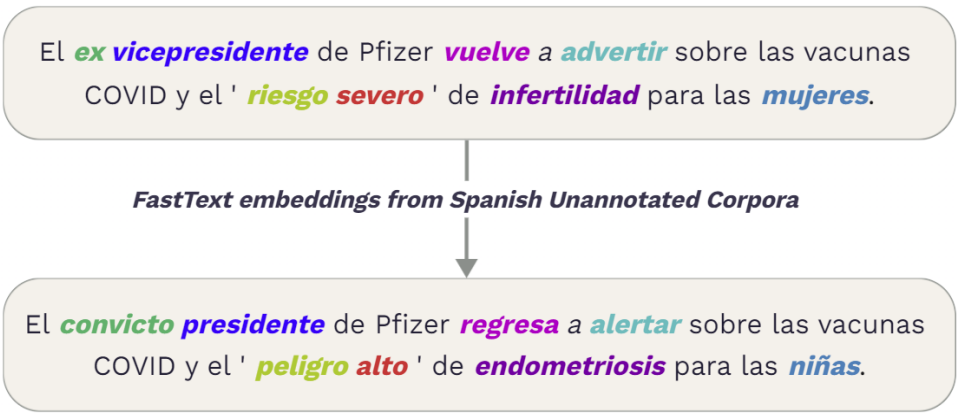


Figure 8.2: Synonym augmented Spanish dataset example

8.2. Data Processing

Data preprocessing and postprocessing are essential for token-classification tasks as they significantly impact model performance and accuracy. Preprocessing involves preparing raw data by applying techniques such as tokenization, stemming, lemmatization, stop word removal, and handling missing values. These steps are critical to ensure that tokens retain the most relevant information while reducing noise, allowing the model to learn more effectively and make precise token-level predictions. Postprocessing, in turn, refines the model's output by converting predicted labels into meaningful formats, correcting potential errors, and ensuring that results adhere to predefined constraints. This ensures the final output is both interpretable and applicable to real-world tasks. Together, these processes help the model train on high-quality data and produce reliable, actionable outputs. The specific preprocessing techniques will be covered in Section 8.2.1, and postprocessing approaches will be discussed in Subsection 8.2.2.

8.2.1. Data Preprocessing

The preprocessing of data was a critical step in the development of an effective token classification model, used for both the Spanish and English datasets. This section outlines the detailed processes undertaken to transform raw text data into a structured format suitable for model training and evaluation.

Tokenization and Label Alignment

The first key step was tokenization, where text was broken down into individual tokens (words or subwords). Each sentence in the dataset was tokenized, and the tokenizer generated word IDs to map tokens back to the original words. During this process,

the alignment of labels with these tokens was meticulously handled to ensure accurate representation of entities. To align the labels, each token was assigned a corresponding label, considering the start and end positions of the annotated entities. Tokens at the beginning of an entity were marked with a 'B-' (beginning) label, while tokens inside an entity received an 'I-' (inside) label. Tokens not part of any entity were labeled as 'O' (outside). This precise alignment was crucial for maintaining the integrity of the annotations throughout the tokenization process.

Enhancement of Dataset with Category Columns

To facilitate better data analysis and stratified sampling, binary indicator columns were added for each label category (CAMPAIGNER, VICTIM, AGENT, FACILITATOR, OBJECTIVE, and NEGATIVE_EFFECT). This involved iterating over the dataset and checking each annotation to determine the presence of specific categories. For each document, a binary column was created for each label, where a value of 1 indicated the presence of the label and 0 indicated its absence. This addition provided a more detailed insight into the distribution of labels across the dataset and was essential for ensuring balanced splits during cross-validation.

Sentence Preparation and Annotation

The preparation of sentences and their annotation involved several steps. Each document was divided into sentences. The sentences were then tokenized, and the positions of tokens within the document were recorded. Annotations were mapped to these tokens based on their character offsets in the original text. For each token in a sentence, the corresponding label was determined by checking the overlap of the token's character range with any annotated spans.

Data Preparation for Model Training

For the model training phase, the data was prepared by splitting it into training and evaluation sets for each fold in a stratified k-fold cross-validation setup. This ensured that each fold had a representative distribution of labels. The prepared sentences, now with aligned tokens and labels, were converted into a format suitable to use with the Hugging Face transformers library[4]. This included creating lists of tokens and their corresponding labels and facilitated efficient data loading and batching during training.

[4] https://huggingface.co/docs/transformers/v4.13.0/index

8.2.2. Data Postprocessing

The postprocessing for this task was different from the binary one. After loading the test dataset into a pandas DataFrame[5], the text was segmented into sentences using spacy's NLP pipeline. The model used for prediction was loaded from a pre-trained directory, and a pipeline was created to handle the token classification task. The pipeline processes the text and generated predictions for each token, including entity labels and confidence scores. However, the predictions were made at the token level, meaning that the raw outputs represented individual tokens rather than entire entities, which may span multiple tokens.

If a sentence exceeded the model's maximum token length, It was truncated to fit within the allowed token length. This is implemented in the code by checking the length of the sentence and slicing it accordingly. The pipeline then made predictions on the truncated text, ensuring that the model could process the input within its constraints. Once the pipeline generated token-level predictions, the next step was to aggregate these predictions into meaningful entity spans. The model's predictions included start and end positions for each token within the sentence, but these positions must be mapped back to the original text. This was done by adjusting the token offsets relative to the start of the sentence. These adjustments ensured that each token's prediction aligns with its position in the original document. The code also handled cases where multiple tokens were part of the same entity. For example, if the entity "COVID - 19 pandemic" was predicted, each word may have received an individual prediction. In the postprocessing step, predictions were merged into a single entity span by aggregating the start and end positions of related tokens and optionally averaging their confidence scores. This aggregation ensures that multi-token entities are represented as single, cohesive spans in the final output.

8.3. Systems

The methodology involves fine-tuning a pre-trained transformer model for token classification, specifically using a BIO tagging scheme for entity recognition. The process begins by setting the environment and configuring parameters for optimal resource usage. Ensuring reproducibility is a priority, and this is achieved by setting a random seed, which guarantees that all processes depending on random operations yield consistent results across different runs. A key component of the approach is adjusting the batch size dynamically according to the available GPU memory. This dynamic adjustment ensures that the training process can handle memory constraints efficiently without causing interruptions, as

[5] `https://pandas.pydata.org/docs/reference/frame.html`

described in Section 6.2. Then all the data preprocessing is done as explained in Section 8.2.1.

During training, a custom learning rate scheduler is employed to adjust the learning rate dynamically over time. For each fold, the model is trained using the HuggingFace Trainer class. Metrics such as precision, recall, F1 score, and accuracy are computed using the seqeval library[6], which is designed for sequence labeling tasks. After training and evaluating the model for all folds, the metrics are averaged across all folds to provide a summary of the model's performance. This averaging step is used solely to gauge the model's overall performance. Finally, the best model checkpoint from cross-validation, based on these performance metrics, is selected and evaluated on the test dataset to provide a final assessment of the model's effectiveness. This cross-validation approach ensures that the model's evaluation metrics are reliable and that the model generalizes well to unseen data. Throughout the process, results are saved to files for further analysis, and necessary directories are created to store the outputs. This methodology ensures that the training process is both efficient and scalable, while also providing robust evaluation through cross-validation.

8.4. Experiments on the Training Dataset

Table 8.1 presents the results for the English dataset. The second column indicates the number of training epochs for each model, with all models trained using data augmentation. The performance of the token classification models in the table reveals notable differences in how each model approaches the task. *roberta-base (15 epochs)* achieves the highest accuracy (0.8897), but its F1 score of 0.5890 indicates a slight imbalance. This model has a better recall (0.6112) than precision (0.5684), meaning it tends to identify more tokens, though it sacrifices precision, suggesting a higher likelihood of false positives. Models such as *distilbert-base-uncased (15 epochs)* and *bert-base-uncased (15 epochs)* follow closely. Their higher recall compared to precision suggests they prioritize identifying more tokens, even if it results in some misclassifications. This trend is common in models focused on maximizing token identification.

[6] `https://pypi.org/project/seqeval/1.2.2/`

Model	Epochs	Accuracy	F1	Precision	Recall
roberta-base	15	**0.8897**	**0.5890**	**0.5684**	**0.6112**
roberta-base	10	0.8210	0.4212	0.3943	0.4520
distilbert-base-uncased	5	0.6932	0.2881	0.2893	0.2869
distilbert-base-uncased	10	0.8457	0.4551	0.4279	0.4860
distilbert-base-uncased	15	0.8563	0.5645	0.5388	0.5928
bert-base-uncased	15	0.8723	0.5278	0.4976	0.5619
dmis-lab/biobert-v1.1	15	0.8439	0.4608	0.4353	0.4894
dslim/bert-base-NER	12	0.7578	0.3412	0.2928	0.4088
xlnet-base-cased	15	0.8856	0.5716	0.5522	0.5924
covid-twitter-bert	4	0.7401	0.3301	0.3279	0.3323

Table 8.1: Token classification models performance metrics for the English dataset

In Table 8.2, we present the results for the Spanish dataset.

As before, the number of training epochs is included in the second column. The performance of the token classification models demonstrates significant differences in how effectively they handle the task. The model *PlanTL-GOB-ES/roberta-base-bne (15 epochs)* stands out with the highest accuracy (0.9326) and a balanced F1 score of 0.5849. Its precision (0.5833) and recall (0.5866) are closely aligned, indicating that it performs consistently in identifying and classifying tokens. The model *Bertin-roberta-base-spanish (12 epochs)* shows similar F1 scores but slightly lower accuracy, with a well-balanced precision and recall. Other models, such as *PlanTL-GOB-ES/roberta-base-bne (8 epochs)* and *bert-base-spanish-wwm-uncased (10 epochs)*, display lower F1 scores, illustrating that fewer training epochs negatively impact their ability to generalize. In contrast, *distilbert-base-spanish-uncased (15 epochs)* offers an intriguing result with a relatively high F1 score (0.5101), showing that compact models can still perform competitively.

Model	Epochs	Accuracy	F1	Precision	Recall
PlanTL-GOB-ES/roberta-base-bne	15	**0.9326**	**0.5849**	**0.5833**	**0.5866**
PlanTL-GOB-ES/roberta-base-bne	8	0.9011	0.4565	0.4562	0.4568
bert-base-spanish-wwm-uncased	10	0.8213	0.4201	0.4093	0.4315
bertin-roberta-base-spanish	12	0.8598	0.5844	0.5856	0.5832
alberto-base-spanish	8	0.7841	0.3841	0.3833	0.3848
distilbert-base-spanish-uncased	15	0.8047	0.5101	0.5019	0.5185
electra-small-spanish	5	0.7288	0.3421	0.3332	0.3513
xlm-roberta-base	5	0.7981	0.4012	0.3834	0.4206

Table 8.2: Token classification models performance metrics for the Spanish dataset

8.5. Submitted Systems for Task 2

In this section, we describe our submitted systems for detecting elements of oppositional narratives (task 2). We approached task 2 by fine-tuning a transformer model, using also strong data processing, described in Section 8.2, and data augmentation, described in Section 8.1. We treated task 2 as a token classification problem by fine-tuning models with a single token classification head. Having only one head (instead of a classification head per label, as implemented in the provided baseline [48]) precluded the possibility of overlapping spans, but offered increased simplicity and reduced computational expense instead. While the provided data was annotated at the document-level, we transformed it so that we could train the token classifier at the sentence-level instead. Segmenting the text into sentences overcame the problem of transformers truncating texts that are longer than the maximum length size, ensuring no data was lost during training or testing.

In the first submitted run, we applied the best model checkpoint without additional training. For the English dataset the *facebook/roberta-base*model was employed while for the Spanish dataset, the *PlanTL-GOB-ES/roberta-base-bne*[7] [40] model was used.

For the second run, the best model checkpoint from the first run was trained for one more epoch using the augmented dataset as train, without having any data for validation. This final model checkpoint was then used to detect the elements of oppositional narratives in the test dataset.

The span-detection techniques described here directly address **RQ2**, exploring the feasibility and challenges of identifying narrative elements in multilingual datasets.

[7] `https://huggingface.co/PlanTL-GOB-ES/roberta-base-bne`

9 | Results on the Test Dataset

The official evaluation metrics were different for each task. For the binary classification task, the Matthews Correlation Coefficient was used as the primary evaluation metric [21], while for the span-level detection task, the macro-averaged span-F1 [24] score was employed. In addition to these primary metrics, binary F1 scores for each class were provided for task 1, and per-category span-F1 scores were reported for task 2.

The shared task organizers provided two hard baseline models for these tasks. For task 1, the baseline is a standard BERT classifier [27]. For task 2, the baseline is a BERT-based multi-task token classifier with separate classification heads and a common transformer backbone.

The baselines utilize either English or Spanish BERT models, depending on the language. The performances of our approaches and the baselines are reported in Table 9.1 (task 1) and in Table 9.2 (task 2). The complete official ranking [48] for the English dataset can be found in Table B.1, while the Spanish official ranking can be found in Table B.2.

	Task 1			
	MCC	**F1-macro**	**F1-conspiracy**	**F1-critical**
English_baseline	*0.7964*	*0.8975*	*0.8632*	*0.9318*
English_submitted_run1	0.7574	0.8769	0.8338	0.9200
English_submitted_run2	0.7872	0.8917	0.8536	0.9297
Spanish_baseline	*0.6681*	*0.8339*	*0.7872*	*0.8806*
Spanish_submitted_run1	0.6147	0.7950	0.7179	0.8720
Spanish_submitted_run2	0.6722	0.8293	0.7699	0.8887

Table 9.1: Performance metrics results for task 1

	Task 2			
	span-P	**span-R**	**span-F1**	**micro-span-F1**
English_baseline	*0.5323*	*0.4684*	*0.6334*	*0.4998*
English_submitted_run1	0.5832	0.6856	0.6293	0.6074
English_submitted_run2	0.5859	0.6790	0.6279	0.6120
Spanish_baseline	*0.4533*	*0.5621*	*0.4934*	*0.4952*
Spanish_submitted_run1	0.5997	0.6193	0.6089	0.6051
Spanish_submitted_run2	0.6159	0.6129	0.6129	0.6108

Table 9.2: Performance metrics results for task 2

9.1. Discussion

This section analyzes the results presented in Tables 9.1 and 9.2. We will provide insights into the results for both English and Spanish datasets, considering the baselines provided by the shared task organizers and how our approaches performed [97]. We organize this discussion by task and language. First, we examine the binary classification results from task 1. We then turn our attention to task 2.

9.1.1. Task 1: Binary Classification

We begin by examining the English dataset results, followed by a focused analysis of the Spanish dataset to assess language-specific performance.

English Results in Task 1

In our analysis of the English dataset, the first submitted run exhibited good performance metrics. The Matthews Correlation Coefficient was 0.7574, indicating a robust ability to distinguish between different narrative types. The F1-macro score of 0.8769 further supported the model's high overall classification capability. Notably, the F1 score for critical thinking texts was 0.9200, compared to 0.8338 for conspiracy texts. This disparity suggests that the model more effectively identified critical thinking, potentially due to the more nuanced nature of conspiracy texts. However, it is important to note that these results are below the baseline, which had an MCC of 0.7964 and an F1-macro score of 0.8975. This baseline is indeed a hard baseline and difficult to beat.

In the second submitted run, the model demonstrated improved reliability, with the MCC rising to 0.7872. The F1-macro score also increased to 0.8917, indicating enhanced overall

performance. The F1 scores for conspiracy and critical texts were 0.8536 and 0.9297, respectively, showing more balanced and accurate classifications. Despite these improvements, the model still did not surpass the baseline, highlighting the baseline's strong performance and the challenges in achieving higher accuracy.

Spanish Results in Task 1

For the Spanish dataset, the first submitted run showed moderate performance with an MCC of 0.6147, indicating the need for further improvement. The F1-macro score was 0.795, reflecting decent overall performance but highlighting areas for enhancement. The model struggled more with conspiratorial texts, achieving an F1 score of 0.7179 for conspiracy versus 0.8720 for critical texts, likely due to the specific linguistic challenges presented by the Spanish language. The baseline for Spanish had an MCC of 0.6681 and an F1-macro score of 0.8339, indicating that the baseline was also strong for this language.

In the second submitted run, there was a noticeable improvement in performance. The MCC increased to 0.6722, and the F1-macro score rose to 0.8293, indicating better overall performance. The F1 scores for conspiracy and critical texts improved significantly to 0.7699 and 0.8887, respectively. These improvements suggest that the model became more adept at handling linguistic features specific to Spanish, leading to more balanced and accurate classifications. This second run beat the strong baseline performance, reflecting significant progress.

9.1.2. Task 2: Span-level Detection

We start with span detection results for English, then we explore the model's effectiveness in identifying narrative elements in Spanish.

English Results in Task 2

In the first submitted run for the English dataset, the model achieved a span-P of 0.5832 and a span-R of 0.6856, indicating moderate precision but better recall. The span-F1 score was 0.6293, and the micro-span-F1 score was 0.6074, suggesting a balanced performance with a need for improvement in precision. The baseline for this task had a span-F1 score of 0.6334 and a micro-span-F1 of 0.4998, showing that our model performed significantly better in terms of micro-span-F1 but slightly underperformed in span-F1 compared to the strong baseline.

In the second submitted run, there was a slight improvement in precision, with a span-P of

0.5859 and a span-R of 0.6790. The span-F1 score remained relatively consistent at 0.6279, and the micro-span-F1 score increased marginally to 0.6120. These modest enhancements reflect steady progress in performance and show that our model maintained competitive performance with the baseline.

Spanish Results in Task 2

For the Spanish dataset, the first submitted run demonstrated a balanced performance, with a span-P of 0.5997 and a span-R of 0.6193. The span-F1 score was 0.6089, and the micro-span-F1 score was 0.6051. The model performed significantly better at span detection in Spanish compared to English, possibly due to distinct narrative markers in the language. The baseline for this task had a span-F1 score of 0.4934 and a micro-span-F1 of 0.4952, indicating that our model significantly outperformed the baseline in both metrics.

In the second submitted run, the performance improved, with a span-P of 0.6159, a span-R of 0.6129, and a span-F1 of 0.6129. These stable scores reflect consistent and reliable performance in detecting narrative elements at the span level, maintaining a significant advantage over the baseline.

Due to the best performances obtained in both English and Spanish for the second task, our model achieved the highest overall scores, leading to the first-place ranking in the shared task competition [97]. Overall, the consistent improvement across both tasks and languages in the second submitted run underscores the effectiveness of the adjustments made in data augmentation, training processes, and model parameter tuning.

9.2. Analysis with and without Data Augmentation

Table 9.3 and Table 9.4 present a comparative analysis of the submitted fine-tuned models with and without data augmentation using the test datasets. For task 1, the results demonstrate that models incorporating data augmentation consistently outperform those without it across all metrics and in both languages. This is especially evident in the MCC and F1-macro scores. For example, in *run1 (English)*, the MCC increased from 0.6880 without augmentation to 0.7574 with it, while the F1-macro score improved from 0.8344 to 0.8769. Notable gains are also observed in the F1-Conspiracy and F1-Critical scores, indicating that data augmentation enhances the model's ability to capture both conspiracy and critical thinking narrative elements more effectively. A similar trend is observed in *run1 (Spanish)*, with the MCC rising from 0.5740 without augmentation to 0.6148 with it. Although the improvement is less pronounced in *run2 (English)* and *run2 (Spanish)*,

data augmentation still results in marginal gains, underscoring its value even in more optimized runs. The performance boost observed for both English and Spanish datasets suggests that the data augmentation strategy is robust across languages. These findings highlight the effectiveness of data augmentation in enhancing span-based performance across different languages.

In Figures 9.1 and 9.2, we can observe that experiments with data augmentation yield better results across all metrics. The dotted lines in these figures represent the experiments without data augmentation, while the solid colored blocks indicate those with data augmentation.

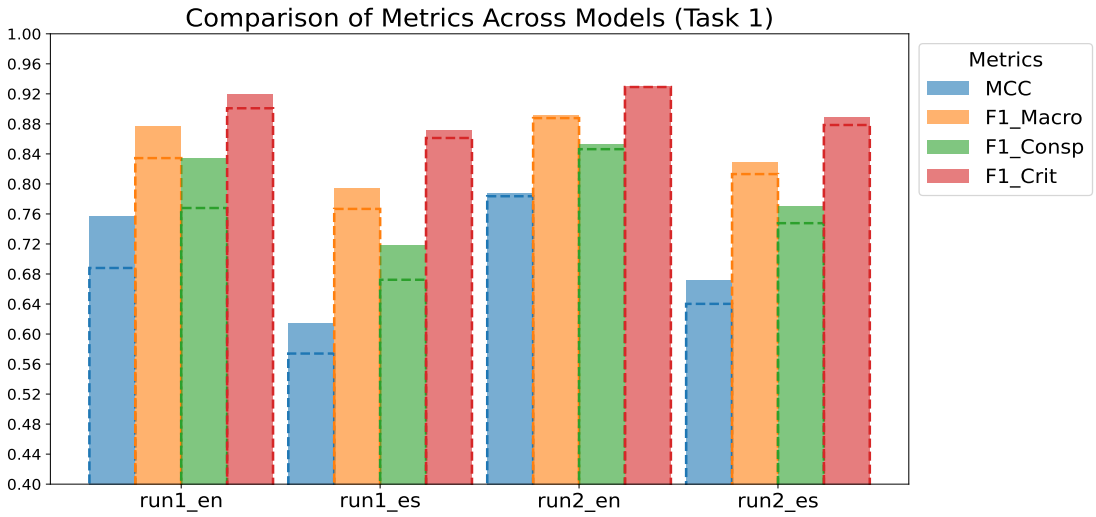


Figure 9.1: Task 1 metrics comparison with and without data augmentation

Model	Data Aug	MCC	F1-Macro	F1-Consp	F1-Crit
ENGLISH					
run1	✓	0.7574	0.8769	0.8338	0.9200
run1		0.6880	0.8344	0.7679	0.9008
run2	✓	0.7872	0.8917	0.8536	0.9297
run2		0.7836	0.8877	0.8463	0.9291
SPANISH					
run1	✓	0.6148	0.7950	0.7179	0.8721
run1		0.5740	0.7667	0.6723	0.8612
run2	✓	0.6722	0.8293	0.7699	0.8887
run2		0.6402	0.8131	0.7477	0.8785

Table 9.3: Evaluation metrics with and without data augmentation for task 1

For task 2, the benefits of data augmentation remain positive, though they are more subtle compared to task 1. For example, in *run1 (English)*, the span-F1 score increases from 0.6180 without augmentation to 0.6404 with it. Similarly, for *run2 (English)*, span-F1

rises from 0.6087 to 0.6279 with data augmentation. The precision and recall values show consistent improvements as well, suggesting that data augmentation enhances the model's ability to correctly identify and classify narrative spans while maintaining a balanced precision-recall tradeoff. In Spanish, the span-F1 improvement is also evident, such as in *run1 (Spanish)*, where it goes from 0.6010 to 0.6215, and in *run2 (Spanish)*, where it increases from 0.5998 to 0.6232.

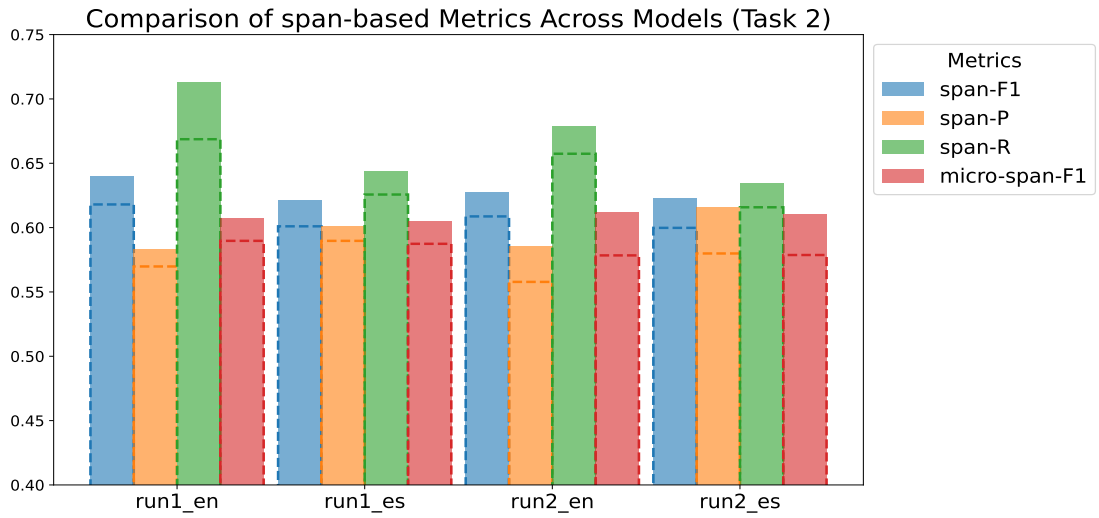


Figure 9.2: Task 2 metrics comparison with and without data augmentation

Model	Data Aug	span-F1	span-P	span-R	micro-span-F1
		ENGLISH			
run1	✓	0.6404	0.5832	0.7133	0.6077
run1		0.6180	0.5698	0.6687	0.5897
run2	✓	0.6279	0.5859	0.6790	0.6120
run2		0.6087	0.5578	0.6574	0.5784
		SPANISH			
run1	✓	0.6215	0.6015	0.6438	0.6051
run1		0.6010	0.5897	0.6257	0.5874
run2	✓	0.6232	0.6159	0.6342	0.6108
run2		0.5998	0.5799	0.6158	0.5787

Table 9.4: Evaluation metrics with and without data augmentation for task 2

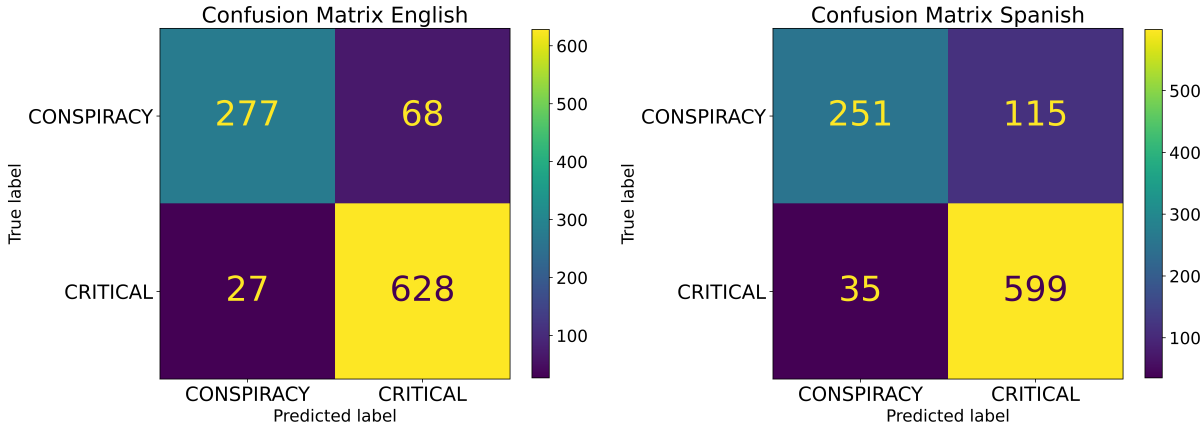
The findings presented in this section provide clear insights into **RQ3**, demonstrating that specific data augmentation techniques, such as back-translation and synonym replacement, have a measurable impact on model performance. This analysis highlights the role of augmentation in improving the model's ability to handle diverse and complex narrative structures across multilingual datasets.

10 | Error Analysis

In this section, we analyze the performance of our submitted models. The error analysis enables us to understand where the model's predictions diverge from the actual categories.

10.1. Task 1: Binary Classification

We begin by examining the confusion matrix of the binary classification task, which provides a visual representation of the model's performance by illustrating the frequency of correct predictions for each category and the frequency of misclassifications.



(a) Confusion matrix for the English dataset (b) Confusion matrix for the Spanish dataset

Figure 10.1: Confusion matrix of binary classification for the English and the Spanish datasets

From the confusion matrix, it is evident that in the Spanish dataset, the model frequently misclassifies critical texts as conspiracy. Specifically, out of 634 critical texts, 35 were misclassified as conspiracy, yielding an error rate of approximately 5.52%. Conversely, of the 366 conspiracy texts, 115 were incorrectly labeled as critical, resulting in a significantly higher error rate of approximately 31.4%. In the English dataset, the pattern differs slightly. Out of 655 critical texts, 27 were misclassified as conspiracy, resulting in an

error rate of approximately 4.12%. On the other hand, out of 345 conspiracy texts, 68 were incorrectly labeled as critical, leading to an error rate of approximately 19.7%. This comparative analysis indicates that while the model generally exhibits greater difficulty in accurately classifying conspiracy texts in both datasets, the Spanish dataset demonstrates a particularly high error rate for misclassifying this category.

Text length can significantly influence classification tasks, as longer texts often introduce greater complexity but at the same time can give more context that the model must interpret. The following plots illustrate the distribution of text lengths for the false positives in both the English and Spanish datasets.

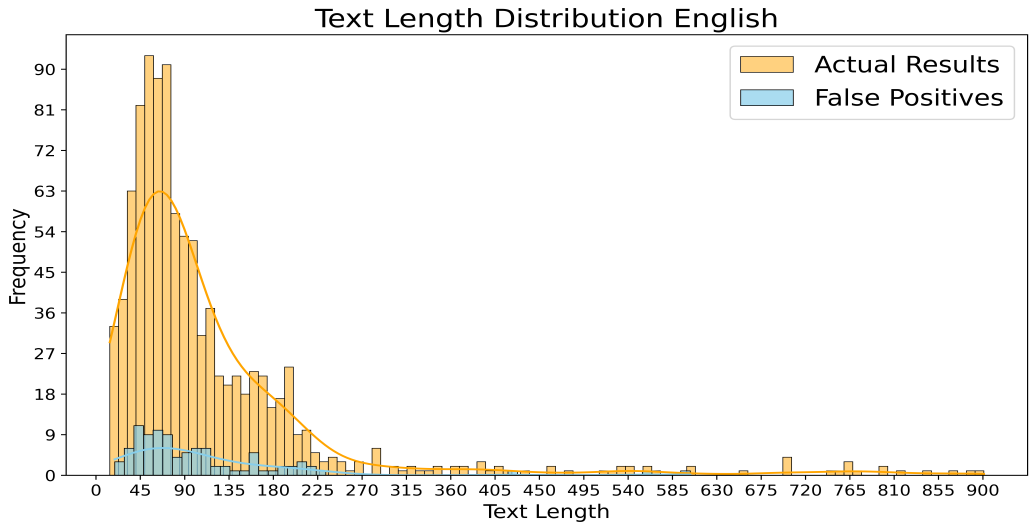


Figure 10.2: Text length distribution in false positives for the English dataset

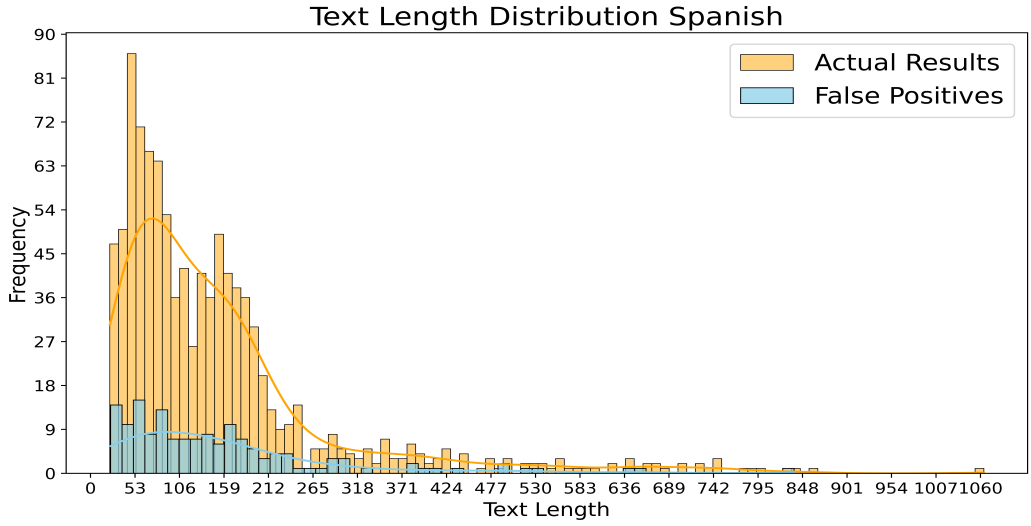


Figure 10.3: Text length distribution in false positives for the Spanish dataset

Figure 10.4: Text length distribution in false positives for the English and Spanish datasets

Analyzing the text lengths of the false positives in the English dataset, we observe that the text lengths range from 19 to 603 words. The average length of the texts in this dataset is approximately 107 words. This indicates a moderate variation in the lengths of false positive texts, with some texts being relatively short while others are significantly longer. In the Spanish dataset, the text lengths of the false positives range from 24 to 838 words, with an average length of 155 words. This dataset exhibits a wider range of text lengths compared to the English dataset, and the average length is also higher. This suggests that false positives in the Spanish dataset tend to have longer texts on average. The analysis indicates that as texts become longer and more detailed, the model's ability to correctly classify them increases. The additional context and complexity in longer texts introduce more opportunities for the model to better classify the data. As we can observe from Figure 10.4 the distribution of the false positives follow the actual results, so we can say that we don't find any unexpected behavior in our system.

The high error rate for conspiracy texts in both datasets might be attributed to several factors. Firstly, ambiguity in language plays a significant role. Many texts contain language that can be interpreted as both critical and conspiratorial, especially when discussing contentious topics such as vaccines or government policies. This overlap makes it challenging for the model to draw clear distinctions. Secondly, the complexity of the context adds to the difficulty. Longer texts provide more context, but they also introduce more opportunities for nuanced or ambiguous statements that the model struggles to classify accurately. The increased detail can obscure the main sentiment or intent of the text. The limitations of the training data also contribute to the misclassifications, If the training data does not adequately cover the range of expressions and contexts found in the test data. Since the train dataset is unbalanced to the critical texts, the model struggles to identify conspiracy theories. Cultural and linguistic nuances further complicate the model's performance. The differences between the English and Spanish datasets might be attributed to cultural and linguistic nuances that influence how critical discussions and conspiracy theories are expressed. These nuances can introduce additional complexity for the model, which needs to be sensitive to the specific contexts and idioms used in each language.

10.2. Task 2: Span-Level Detection

Continuing from the binary classification task, we now turn our attention to the more intricate domain of span-level detection. This analysis concentrates on the diverse categories within span text to identify discrepancies between the model's predictions and the

actual annotations, by examining the confusion matrices for both the English and Spanish datasets. These matrices visually represent the model's performance, illustrating the frequency of correct predictions for each category and the frequency of misclassifications.

(a) Confusion matrix for the English dataset (b) Confusion matrix for the Spanish dataset

Figure 10.5: Confusion matrix of span-text categories for the English and the Spanish datasets

The confusion matrices reveal several notable patterns. For the English dataset, the model correctly identified 858 spans as *Agent* but also mislabeled 342 spans that have been not annotated. In the *Campaigner* category, the model correctly identified 882 spans but misclassified 325 spans that were not annotated. The *Facilitator* category had 308 correct identifications and 229 misclassifications into *No_Annotation*. The *Negative_Effect* category had 782 correct identifications but also saw 269 misclassifications. The *Objective* category had 218 correct identifications but 175 misclassifications, and the *Victim* category had 566 correct identifications with 239 misclassifications.

A significant issue is observed with *No_Annotation*, where spans that should not have been annotated were misclassified into various categories: 498 as *Agent*, 399 as *Campaigner*, 327 as *Facilitator*, 820 as *Negative_Effect*, 298 as *Objective*, and 375 as *Victim*. This broad issue suggests difficulty in identifying when no annotation is needed.

For the Spanish dataset, the confusion matrix reveals similar challenges. The model correctly identified 308 spans as *Agent* but misclassified 271 spans as *No_Annotation*. In the *Campaigner* category, 584 spans were correctly identified, but 316 spans were misclassified. The *Facilitator* category had 248 correct identifications with 211 misclassifications. The *Negative_Effect* category had 930 correct identifications but 351 misclassi-

fications.The *Objective* category had 50 correct identifications with 65 misclassifications. The *Victim* category had 647 correct identifications but 337 spans were misclassified as *No_Annotation*.

The Spanish model also frequently misclassified *No_Annotation* spans: 198 as *Agent*, 259 as *Campaigner*, 213 as *Facilitator*, 658 as *Negative_Effect*, 79 as *Objective*, and 299 as *Victim*.

Comparing the English and Spanish datasets, both exhibit high misclassification rates in the *No_Annotation* category, indicating challenges in identifying spans that should not be annotated. This issue likely stems from the complexity and varied contexts in which terms appear.

This discrepancy can be attributed to several factors. Model overconfidence may lead to identifying spans based on patterns learned from the training data. This overconfidence might derive from the model's inability to accurately gauge the certainty of its predictions, leading to an excess of false positives. Annotation inconsistencies could also play a role, as human annotators might miss certain spans, especially in lengthy or complex texts, leading to discrepancies between the predicted and actual spans.

The model's contextual sensitivity might also contribute to this issue, as it may be picking up on contextual cues suggesting the presence of relevant spans, even if they were not explicitly annotated. This could indicate the model's sensitivity to linguistic or contextual markers not consistently labeled in the training data. For example, if certain phrases or contexts are typically associated with a specific category in the training data, the model might overgeneralize these associations to the test data.

11 | Experiments with LLMs and Multilingual Transformer Models

In this chapter, we explore further experiments conducted after submitting our runs to the shared task. These additional tests aimed to evaluate the effectiveness of both multilingual transformer models and Large Language Models (LLMs). We began by fine-tuning both model types to tailor their predictions, while for LLMs specifically, we also investigated zero-shot learning *(ZSL)* and few-shot learning *(FSL)* scenarios. The objective was to determine whether these approaches could outperform our submitted runs and yield more robust results across the two datasets. We will start with the first task and then continue with the second one.

11.1. Experiments Task 1

This section addresses **RQ1**, examining the effectiveness of multilingual models and Large Language Models in distinguishing between conspiratorial and critical thinking narratives. We tested various configurations to enhance performance, including fine-tuning multilingual models on both English and Spanish datasets to leverage cross-linguistic strengths. We further evaluated LLMs using zero-shot and few-shot learning to assess their capacity for robust classification without explicit training, as well as the impact of fine-tuning methods incorporating task-specific prompts for optimized results.

11.1.1. Multilingual Transformer Models

The experiments with multilingual transformer models aimed to leverage the capability of handling both English and Spanish texts within a single model. Unlike the earlier approaches where the datasets were translated for consistency, these models were fine-tuned directly on the original language texts using appropriate spaCy models for language-

specific embeddings. This approach allowed us to maintain the linguistic richness of each language while enabling the models to process bilingual datasets effectively.

The results of these experiments are shown in Table 11.1. We also evaluated ensemble models that combined predictions from the three single multilingual transformers to enhance performance. These ensembles aimed to leverage each model's strengths for more robust predictions. The submitted runs to the shared task are also included in the table for comparison.

Model	Acc.	MCC	Prec.	Recall	F1-Consp	F1-Crit	F1-Macro
ENGLISH							
xlm-roberta-large	**0.8980**	**0.7715**	0.9147	**0.7768**	**0.8401**	**0.9251**	**0.8826**
bert-base-multilingual	0.8760	0.7237	**0.9300**	0.6928	0.7940	0.9113	0.8527
xlm-roberta-base	0.8750	0.7191	0.9015	0.7159	0.7981	0.9095	0.8538
ensemble-3_multilingual	0.8900	0.7555	0.9401	0.7275	0.8203	0.9207	0.8705
submitted_run	*0.9050*	*0.7872*	*0.9112*	*0.8029*	*0.8536*	*0.9297*	*0.8917*
SPANISH							
xlm-roberta-large	**0.8180**	**0.6027**	0.8770	**0.5847**	**0.7016**	**0.8691**	**0.7854**
bert-base-multilingual	0.7730	0.5011	0.8564	0.4563	0.5954	0.8423	0.7188
xlm-roberta-base	0.7970	0.5632	**0.9137**	0.4918	0.6394	0.8587	0.7491
ensemble-3_multilingual	0.8010	0.5687	0.8957	0.5164	0.6551	0.8602	0.7576
submitted_run	*0.8500*	*0.6722*	*0.8776*	*0.6858*	*0.7699*	*0.8887*	*0.8293*

Table 11.1: Performance metrics of multilingual transformer models for binary classification

While the multilingual transformer models demonstrated promising results, they generally performed slightly worse compared to the submitted runs. For instance, the best-performing multilingual transformer model, *xlm-roberta-large*[1], achieved an accuracy of 0.8980 and an MCC of 0.7715 on the English dataset, which, while strong, did not surpass the submitted run's accuracy of 0.9050 and MCC of 0.7872. Similarly, for the Spanish dataset, the submitted run achieved a higher accuracy (0.8500) and MCC (0.6722) compared to the top-performing multilingual model (0.8180 accuracy and 0.6027 MCC).

These results highlight that although the multilingual transformer models are capable of handling both languages within a single framework, the initial models submitted offered a slightly better fit for their respective languages. One possible reason could be that we provided the model with both the original dataset and the translated one, resulting in an equal amount of data for both monolingual and multilingual models. Nonetheless, the

[1] `https://huggingface.co/FacebookAI/xlm-roberta-large`

multilingual approach remains valuable for its efficiency in managing bilingual data and reducing the need for separate models, offering a streamlined alternative for multilingual tasks.

11.1.2. Zero Shot Learning

Taking a further step to the most cutting-edge technologies, this section outlines the methodologies used leveraging Large Language Models, prompt-based generation, and optimized data handling to ensure efficient and scalable performance. These experiments were conducted to evaluate whether LLMs, even though not designed explicitly for binary classification, could achieve better results than specialized fine-tuned transformer based models. Various LLMs were selected for their ability to handle instruction-based tasks effectively.

These models were loaded using half-precision floating-point numbers to reduce memory consumption while maintaining computational efficiency. Additionally, was included an option for 4-bit quantization to further reduce the memory footprint, making it feasible to run these models on more limited hardware resources. The classification system was powered by a text-generation pipeline, built using Hugging Face's transformer libraries. This pipeline handled the input texts and generated deterministic responses. The output length was capped at 80 tokens to keep the generated responses concise and focused on the classification task.

At the core of the classification process was a structured prompt in the language of each dataset, guiding the model to categorize texts into one of the two categories. For zero-shot learning, the Spanish dataset was prompted in Spanish, while the English dataset used an English prompt. The prompt used for the English dataset was as follows:

```
 prompt = ‘‘<s>[INST] «SYS»
You are an expert critical thinker specialized in analyzing public health narratives,
particularly regarding the COVID-19 pandemic.You are tasked with classifying a text
into one of these two categories:
1.  CRITICAL THINKING: Texts that question major public health decisions but do not
suggest that secret, powerful, or malevolent groups are behind these decisions.
2.  CONSPIRACY THEORIES: Texts that imply public health decisions, especially those
regarding the COVID-19 pandemic, are part of a plot orchestrated by secret, powerful,
and malevolent groups.
Now, say if the following text is ’CRITICAL’ or ’CONSPIRACY’:
«/SYS»
{user_text}
[/INST]</s> ’’
```

Model	Acc.	MCC	F1-macro	F1-Consp.	F1-Crit.
			ENGLISH		
Llama-3.2-3B-Instruct	0.5010	0.1124	0.5010	0.5035	0.4985
gemma-2-9b-it	0.6400	**0.4602**	**0.6394**	**0.6545**	0.6245
Mistral-7B-Instruct	0.3670	0.0921	0.2953	0.5201	0.0705
gemma-7b-it	**0.6500**	-0.0246	0.3993	0.0113	**0.7874**
submitted_run	*0.9050*	*0.7872*	*0.8917*	*0.8536*	*0.9297*
			SPANISH		
Llama-3.2-3B-Instruct	**0.6470**	0.1260	0.4671	0.1575	**0.7767**
gemma-2-9b-it	0.5600	**0.3265**	**0.5517**	**0.6127**	0.4907
Mistral-7B-Instruct	0.4800	0.0761	0.4777	0.5122	0.4433
gemma-7b-it	0.3730	0.0099	0.2883	0.5338	0.0427
submitted_run	*0.8500*	*0.6722*	*0.8293*	*0.7699*	*0.8887*

Table 11.2: Performance metrics of zero-shot learning for binary classification

Analyzing the results in Table 11.2, it becomes evident that the zero-shot learning achieved significantly lower performance compared to the submitted systems. The highest accuracy achieved by the zero-shot LLMs was 0.6500 with *gemma-7b-it*[2] model on the English dataset, which is considerably lower than the 0.9050 accuracy of the submitted English system. The highest MCC among the zero-shot models was 0.4602, achieved by *gemma-2-9b-it*[3] model on the English dataset, which is also substantially lower than the MCC of 0.7872 achieved by the submitted system. This underlines that the zero-shot learning reached very poor performance compared to the submitted systems, indicating that without fine-tuning, LLMs struggle to match the effectiveness of models specifically trained for this task.

The suboptimal performance of zero-shot learning in the binary classification task likely stems from several factors. One key issue is the nuanced difference between Critical and Conspiracy texts, which may be too subtle for an LLM to distinguish without fine-tuning. The distinctions in language used across these categories often involve tone or implied intentions, which LLMs might not capture with generic knowledge and prompts alone. Consequently, the models may have failed to differentiate adequately between texts that question public health decisions and those implying a hidden agenda, leading to misclassifications.

Another reason could be that the LLMs did not fully grasp the classification task despite the structured prompt. The complex, instruction-based prompt may not have been enough to guide the model effectively, as zero-shot learning relies heavily on how well the task

[2] `https://huggingface.co/google/gemma-7b-it`
[3] `https://huggingface.co/google/gemma-2-9b-it`

aligns with the model's existing knowledge. Furthermore, without task-specific training, the LLMs were prone to overgeneralization or an inadequate representation of the decision boundaries between the two categories.

11.1.3. Few Shot Learning

We incorporated few-shot learning into our experiments to assess whether including examples in the prompts would enhance the models' classification accuracy compared to the zero-shot approach.

By providing five examples per category, we aimed to give the models a clearer understanding of how to differentiate between the two categories. This approach aligns with recent research emphasizing the importance of guiding large language models through well-defined prompts to improve performance in text classification tasks [75]. Also in this case each dataset was prompted in its respective language.

The complete performance results are presented in Table 11.3, and the prompt in English used for these experiments is detailed below.

```
 prompt = ‘‘<s>[INST] «SYS»
You are an expert critical thinker specialized in analyzing public health narratives,
particularly regarding the COVID-19 pandemic.  You are tasked with classifying a text
into one of these two categories:
1.  CRITICAL THINKING: Texts that question major public health decisions but do not
suggest that secret, powerful, or malevolent groups are behind these decisions.
2.  CONSPIRACY THEORIES: Texts that imply public health decisions, especially those
regarding the COVID-19 pandemic, are part of a plot orchestrated by secret, powerful,
and malevolent groups.
Here below there are some examples:
--- Start of the Examples ---
<EXAMPLES>
--- End of the Examples ---
Now, say if the following text is ’CRITICAL’ or ’CONSPIRACY’:
«/SYS»
{user_text}
[/INST]</s> ’’
```

Model	Acc.	MCC	F1-macro	F1-Cons.	F1-Crit.
ENGLISH					
Llama-3.2-3B-Instruct	**0.6970**	0.2559	0.5635	0.3221	**0.8049**
gemma-2-9b-it	0.6870	**0.5049**	**0.6869**	**0.6822**	0.6916
Mistral-7B-Instruct	0.5090	0.2164	0.5039	0.5540	0.4538
gemma-7b-it	0.5410	0.1208	0.5361	0.4883	0.5839
submitted_run	*0.9050*	*0.7872*	*0.8917*	*0.8536*	*0.9297*
SPANISH					
Llama-3.2-3B-Instruct	**0.6480**	0.1315	0.4750	0.1737	**0.7764**
gemma-2-9b-it	0.5910	**0.3322**	**0.5859**	**0.6181**	0.5597
Mistral-7B-Instruct	0.6410	0.0914	0.4337	0.0911	0.7763
gemma-7b-it	0.6440	0.1158	0.4744	0.1759	0.7730
submitted_run	*0.8500*	*0.6722*	*0.8293*	*0.7699*	*0.8887*

Table 11.3: Performance metrics of few-shot learning for binary classification

Despite including few-shot examples, the performance improvements were marginal and still significantly lower than the submitted systems. For example, the *gemma-2-9b-it* model on the English dataset achieved an accuracy of 0.6870 and an MCC of 0.5049, which is still substantially lower than the submitted system's accuracy of 0.9050 and MCC of 0.7872. This further emphasizes that even with few-shot learning, the LLMs reached very poor performance compared to the submitted systems.

11.1.4. Zero and Few Shot Learning with the Text Classification Pipeline

We also explored using the LLM models in *"text-classification"* mode instead of *"text-generation"* as previously employed. This classification pipeline adapted LLMs to return one of the two predefined categories directly from the input text. By streamlining the process, the system avoided generating extra tokens or context, ensuring faster execution and consistent binary labeling. Each dataset entry was processed and classified efficiently, with results recorded alongside the corresponding text and identifier. Unlike the text-generation approach, this method focused solely on categorizing input text rather than generating new content. The performance metrics for zero-shot and few-shot learning in this setup are presented in Table 11.4.

Model	Acc.	MCC	F1-macro	F1-Consp.	F1-Crit.
ZERO SHOT					
English					
Llama-3.2-3B-Instruct	0.4130	**0.0866**	0.3846	**0.5169**	0.2522
gemma-2-9b-it	0.4060	0.0683	0.3764	0.5123	0.2404
Mistral-7B-Instruct	**0.6570**	0.0526	**0.4074**	0.0228	**0.7920**
gemma-7b-it	0.6500	-0.0246	0.3993	0.0113	0.7874
Spanish					
Llama-3.2-3B-Instruct	0.3920	**0.0464**	0.3283	**0.5352**	0.1214
gemma-2-9b-it	0.3740	-0.2007	0.3739	0.3651	0.3826
Mistral-7B-Instruct	**0.6563**	0.0000	**0.3962**	0.0000	**0.7924**
gemma-7b-it	0.3730	0.0099	0.2883	0.5338	0.0427
FEW SHOT					
English					
Llama-3.2-3B-Instruct	0.5260	**-0.0127**	**0.4922**	0.3612	0.6232
gemma-2-9b-it	0.6320	-0.0813	0.3973	0.0213	0.7734
Mistral-7B-Instruct	0.3590	-0.0296	0.3057	**0.4980**	0.1134
gemma-7b-it	**0.6540**	-0.0230	0.3954	0.0001	**0.7908**
Spanish					
Llama-3.2-3B-Instruct	0.4920	**-0.0481**	**0.4730**	0.3728	0.5731
gemma-2-9b-it	**0.4940**	-0.0624	0.4678	0.3496	**0.5859**
Mistral-7B-Instruct	0.4250	-0.1420	0.4181	0.3547	0.4815
gemma-7b-it	0.3650	-0.0557	0.2847	**0.5243**	0.0451
submitted_run English	*0.9050*	*0.7872*	*0.8917*	*0.8536*	*0.9297*
submitted_run Spanish	*0.8500*	*0.6722*	*0.8293*	*0.7699*	*0.8887*

Table 11.4: Zero and few shot learning in text-classification mode for binary classification

The results show that the submitted systems, fine-tuned for the binary classification task, also in this case, outperformed the general-purpose LLMs. Their higher accuracy and MCC values emphasize the value of task-specific fine-tuning for better accuracy and reliability. These results indicate that while LLMs can handle binary classification tasks, specialized transformers fine-tuned on task-specific data are more effective. The poor performance of zero-shot and few-shot LLMs compared to the submitted systems underscores the need for fine-tuning in complex classification tasks.

11.1.5. Fine-Tuning LLMs

Recognizing the potentials of LLMs, we fine-tuned them on the task-specific train data, in order to see if the results were better than the approaches with transformers models and the zero and few shot learning methods.

Fine-tuning Large Language Models involved adapting text-generation models to new tasks, allowing them to specialize in distinguishing between the two categories, using

techniques like quantization and Parameter-Efficient Fine-Tuning (PEFT) to optimize performance while managing computational resources.

Quantization was employed to reduce the model's memory footprint by storing weights in a 4-bit format using NormalFloat4 (NF4) precision, complemented by BF16 for computations. This technique allowed the model to retain its processing capabilities while significantly reducing memory usage. To further enhance efficiency, Low-Rank Adaptation (LoRA) [43] was used as a PEFT method. LoRA fine-tuned only a small subset of the model's parameters, specifically those involved in attention projections, while keeping most of the pre-trained weights frozen.

This approach enabled the model to adapt to the new classification task without the need for extensive parameter updates, thus minimizing computational demands. Training was configured with small batch sizes, augmented through gradient accumulation, and optimized using an 8-bit compatible optimizer, ensuring effective learning despite the model's size.

Model	**Acc.**	**MCC**	**F1-Cons**	**F1-Crit**	**F1-macro**
			ENGLISH		
LLAMA 3.2-3B_NoAug	0.9140	0.8077	0.8677	0.9363	0.9020
LLAMA 3.2-3B_DataAug	0.9110	0.8009	0.8624	0.9342	0.8983
LLAMA 3.2-3B-Instruct	0.8930	0.7632	0.8243	0.9231	0.8737
gemma-2-9b-it	0.9170	0.8144	0.8744	0.9380	0.9062
LLAMA 3.1-8B	0.9050	0.8037	0.8728	0.9242	0.8985
ensemble_3_LLMs	0.9210	0.8236	0.8819	0.9406	0.9113
ensemble_4_LLMs	**0.9300**	**0.8477**	**0.9011**	**0.9458**	**0.9235**
submitted_run	*0.9050*	*0.7872*	*0.8536*	*0.9297*	*0.8917*
			SPANISH		
LLAMA 3.2-3B_NoAug	0.8020	0.5770	0.7346	0.8421	0.7883
LLAMA 3.2-3B_DataAug	0.8260	0.6234	0.7119	0.8754	0.7936
LLAMA 3.2-3B-Instruct	0.8290	0.6268	0.7255	0.8758	0.8007
gemma-2-9b-it	0.8510	0.6737	0.7759	0.8884	0.8322
LLAMA 3.1-8B	0.8620	0.7003	0.8073	0.8925	0.8499
ensemble_3_LLMs	0.8540	0.6810	0.7774	0.8914	0.8344
ensemble_4_LLMs	**0.8870**	**0.7538**	0.8355	**0.9139**	**0.8747**
submitted_run	*0.8500*	*0.6722*	*0.7699*	*0.8887*	*0.8293*

Table 11.5: Performance metrics of fine-tuning LLMs for binary classification

Despite not using the full computational power of the LLMs, this approach resulted in significant performance gains. As presented in Table 11.5, fine-tuned models such as *gemma-2-9b-it* model for the English dataset achieved an accuracy of 0.9170 and an MCC of 0.8144, surpassing the metrics of the submitted English run. The ensemble models *(Llama 3.2-3B-Instruct, gemma-2-9b-it, Llama 3.1-8B and Llama-3.2-3B)* achieved even

higher results, with an accuracy of 0.9300 and an MCC of 0.8477. In the case of the Spanish dataset, fine-tuning led to similar improvements, with the best model reaching an accuracy of 0.8870 and an MCC of 0.7538. While these results outperformed the submitted Spanish run's metrics (0.8500 accuracy and 0.6722 MCC), they demonstrated the benefits of adapting LLMs through fine-tuning.

Moreover, fine-tuning an LLM, even with optimizations like quantization that limit the model's full power, not only outperformed out submitted systems but also achieved the best results in the official ranking of the shared task. The complete official ranking of the shared task can be found in Appendix B [48]. This indicates that the specialized adaptation of LLMs through fine-tuning remains a powerful approach, providing state-of-the-art performance in binary classification tasks.

11.1.6. Fine-Tuning LLMs with Zero and Few Shot Learning

In this section, we extended traditional fine-tuning of LLMs by incorporating a structured prompt into the training process. Our goal was to determine whether adding a task-specific prompt could enhance the model's performance or help it converge more quickly to the results achieved with standard fine-tuning. To maintain consistency and gain meaningful insights, we used the *Llama-3.2-3B-Instruct*[4] model across all experiments, testing two distinct approaches.

The first approach involved using a task-specific prompt integrated directly into the input data during preprocessing. This prompt described the classification task clearly, ensuring that each text input was prefixed with detailed instructions for the model. The prompt was tokenized together with the input texts.

The second approach added a twist by dynamically integrating the task-specific context directly during model execution. Instead of embedding the prompt during preprocessing, we employed a custom *PromptContextTrainer* class to concatenate the prompt with each input batch at runtime. This meant that the model always received the instruction in real-time, ensuring a consistent understanding of the task context. We modified the loss computation to include this added context, keeping the model well-informed of the classification goals.

The prompt used for the zero shot learning in English was the following:

```
prompt = ‘‘<s>[INST] You are tasked with classifying whether a given text is a
CRITICAL statement questioning public health decisions, or a CONSPIRACY narrative
alleging a secret plot related to the pandemic.  [/INST]</s> ’’
```

[4] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

The performance metrics summarized in Table 11.6 show that traditional fine-tuning without prompts, labeled *No_ZSL*, was still the best method. This demonstrates that a straightforward approach effectively utilized the models' pre-trained knowledge for binary classification. However, incorporating structured prompts had mixed effects. The first method, which embedded prompts during preprocessing (*ZSL_preproc*), led to a decline in performance, with the English model's accuracy dropping to 0.9050 and an even more substantial decrease for the Spanish model. This suggests that integrating prompts at the preprocessing stage may have disrupted the models' understanding and introduced unnecessary complexity. In contrast, the second method, which dynamically integrated prompts during training (*ZSL_train_ctx*), produced results closer to the baseline, with only minor differences in accuracy and MCC. Still, even this approach did not surpass the performance of traditional fine-tuning, indicating that the added task-specific prompts offered limited benefits.

Model	Acc.	MCC	F1-Cons	F1-Crit	F1-Macro
ENGLISH					
Llama-3.2-3B_No_ZSL	**0.9110**	**0.8009**	**0.8624**	**0.9342**	**0.8983**
Llama-3.2-3B_ZSL_preproc	0.9050	0.7883	0.8494	0.9306	0.8900
Llama-3.2-3B_ZSL_train_ctx	0.9100	0.7987	0.8649	0.9325	0.8987
SPANISH					
Llama-3.2-3B_No_ZSL	**0.8830**	**0.7465**	**0.8377**	**0.9085**	**0.8731**
Llama-3.2-3B_ZSL_preproc	0.8520	0.6760	0.7849	0.8872	0.8360
Llama-3.2-3B_ZSL_train_ctx	0.8710	0.7196	0.8190	0.8998	0.8594

Table 11.6: Performance metrics for fine-tuned models with ZSL for binary classification

Given that the zero-shot learning approach did not produce performance improvements, we explored a few-shot learning strategy to see if adding explicit examples to the prompts would yield better results. In this setup, we included five sample texts for critical and conspiracy categories, aiming to give the model a clearer understanding of the classification task. The prompt used is shown below:

```
 prompt = ‘‘<s>[INST] You are tasked with classifying whether a given text is a
CRITICAL statement questioning public health decisions, or a CONSPIRACY narrative
alleging a secret plot related to the pandemic.
Here below there are some examples:
--- Start of the Examples ---
- Start Examples of CRITICAL texts -
<EXAMPLES_CRITICAL>
- End Examples of CRITICAL texts -
```

```
- Start Examples of CONSPIRACY texts -
<EXAMPLES_CONSPIRACY>
- End Examples of CONSPIRACY texts -
--- End of the Examples ---
[/INST]</s> ''
```

As shown in Table 11.7, the few-shot learning results fell short compared to the baseline models fine-tuned without prompts. In the first approach, where examples were incorporated into the prompt during preprocessing, performance declined. The English model's accuracy dropped to 0.8840, with an MCC of 0.7401, while the Spanish model's accuracy decreased to 0.8330, with an MCC of 0.6291. This marked a clear regression from the strong performance of traditional fine-tuning, suggesting that embedding examples during preprocessing introduced unnecessary complexity instead of enhancing clarity.

The second approach, which dynamically added examples during training, produced slightly better results but still failed to match the baseline. The English model achieved 89.10% accuracy with an MCC of 0.7552, and the Spanish model showed modest improvement but continued to lag behind the original fine-tuned models. Even with more refined integration, the examples did not elevate the performance beyond that of straightforward fine-tuning.

Overall, the few-shot learning strategy, despite incorporating explicit examples, did not improve the model performance. The baseline models using standard fine-tuning without prompts remained superior. These findings indicate that the added complexity of few-shot prompts may have caused confusion rather than delivering the intended benefits.

Model	Acc.	MCC	F1-Cons	F1-Crit	F1-Macro
ENGLISH					
Llama-3.2-3B_No_FSL	**0.9110**	**0.8009**	**0.8624**	**0.9342**	**0.8983**
Llama-3.2-3B_FSL_preproc	0.8840	0.7401	0.8135	0.9158	0.8647
Llama-3.2-3B_FSL_train_ctx	0.8910	0.7552	0.8305	0.9197	0.8751
SPANISH					
Llama-3.2-3B_No_FSL	**0.8830**	**0.7465**	**0.8377**	**0.9085**	**0.8731**
Llama-3.2-3B_FSL_preproc	0.8330	0.6291	0.6888	0.7756	0.8581
Llama-3.2-3B_FSL_train_ctx	0.8520	0.6761	0.7846	0.8870	0.8698

Table 11.7: Performance metrics for fine-tuned models with FSL for binary classification

11.2. Experiments Task 2

Shifting to the second task, this section tackles **RQ2** by detailing experiments centered on two main strategies: employing multilingual transformer models and exploring zero-shot and few-shot learning with Large Language Models.

Subsection 11.2.1 describes our use of fine-tuned multilingual transformer models to handle both English and Spanish datasets seamlessly, leveraging the linguistic capabilities of spaCy to retain nuanced language features, while Subsection 11.2.2 examines the application of zero-shot and few-shot learning with LLMs, highlighting the impact of prompt designs and the inclusion of example texts on model performance.

11.2.1. Multilingual Transformer Models

Also for the further experiments on the second task, we made use of fine-tuning multilingual transformer models. This method changes from the one described in Section 8.3, by enabling multilingual processing to allow a single model to handle datasets in both English and Spanish. This approach leverages multilingual transformer models that can be fine-tuned on both English and Spanish datasets simultaneously, streamlining the training process and enabling seamless predictions across languages.

In addition, there was not necessary to augment the dataset via synonym replacement because we could use the two datasets available together. Indeed, a key aspect of this method is the use of language-specific spaCy models during preprocessing. This ensures that the linguistic features and nuances of each language are accurately captured during tokenization and text processing. The appropriate spaCy model is dynamically selected based on the language of each text. During training, the model learns to identify and classify entities across languages using the BIO tagging scheme. This bilingual capability means that after training, the same model can make predictions on both English and Spanish datasets without needing any modifications or separate models. This ensures consistency in entity recognition across different languages, making it especially useful for multilingual applications.

In Table 11.8 we can observe the performence metrics on the test dataset. The last word indicate the aggregation strategy used to construct the predictions, and the presence of the checkmark in the *"AddEpoch"* column underlines if the best model checkpoint was retrained for an additional epoch using all the augmented dataset as training. The results demonstrate that the multilingual approach using *xlm-roberta-large* with the additional retrain and the *max* aggregation startegy achieves better results than the *submitted runs*

for both English and Spanish datasets. The micro-span-F1 metric, which is the main one used in the shared task, shows a clear improvement with this approach. Specifically, the *xlm-roberta-large* model reached a micro-span-F1 of 0.6245 for English and 0.6331 for Spanish, surpassing the scores of 0.6120 and 0.6108 achieved by the respective submitted runs.

Model	AddEpoch	span-F1	span-P	span-R	micro-span-F1
		ENGLISH			
xlm-roberta-base_first		0.6002	0.5400	0.6827	0.5843
xlm-roberta-base_max		0.6021	0.5421	0.6841	0.5862
xlm-roberta-base_first	✓	0.6005	0.5488	0.6682	0.5846
xlm-roberta-base_max	✓	0.6013	0.5494	0.6693	0.5854
bert-base-multilingual_first		0.5691	0.5337	0.6118	0.5532
bert-base-multilingual_max		0.5699	0.5345	0.6125	0.5540
bert-base-multilingual_first	✓	0.5859	0.5325	0.6570	0.5700
bert-base-multilingual_max	✓	0.5857	0.5327	0.6562	0.5698
xlm-roberta-large_first		0.6191	0.5835	0.6609	0.6032
xlm-roberta-large_max		0.6191	0.5840	0.6602	0.6032
xlm-roberta-large_first	✓	0.6264	0.5913	0.6673	0.6105
xlm-roberta-large_max	✓	**0.6372**	**0.5918**	**0.6885**	**0.6245**
submitted_run	✓	*0.6279*	*0.5859*	*0.6790*	*0.6120*
		SPANISH			
xlm-roberta-base_first		0.6041	0.5483	0.6786	0.5882
xlm-roberta-base_max		0.6050	0.5490	0.6797	0.5891
xlm-roberta-base_first	✓	0.6020	0.5563	0.6596	0.5861
xlm-roberta-base_max	✓	0.6039	0.5587	0.6608	0.5880
bert-base-multilingual_first		0.5651	0.5384	0.5952	0.5492
bert-base-multilingual_max		0.5665	0.5416	0.5940	0.5506
bert-base-multilingual_first	✓	0.5793	0.5434	0.6226	0.5634
bert-base-multilingual_max	✓	0.5801	0.5439	0.6236	0.5642
xlm-roberta-large_first		0.6330	0.6093	0.6581	0.6171
xlm-roberta-large_max		0.6341	**0.6113**	0.6580	0.6182
xlm-roberta-large_first	✓	0.6355	0.6047	0.6702	0.6196
xlm-roberta-large_max	✓	**0.6358**	0.6049	**0.6707**	**0.6331**
submitted_run	✓	*0.6129*	*0.6199*	*0.6129*	*0.6108*

Table 11.8: Performance metrics of multilingual transformer models for span-text detection

These superior results can be attributed to the models having access to twice as much data. Unlike the submitted systems, which were trained on a single-language dataset and an augmented version using synonym replacement, the multilingual models benefitted from using both language datasets in their entirety. This enriched training data provided better generalization and improved the models' ability to identify spans accurately across different languages. The higher span-F1, span-P, and span-R metrics further underscore the models' enhanced precision and recall in identifying relevant spans.

11.2.2. Zero and Few Shot Learning

Also for the token-classification task we tried to leverage the power of LLMs by firstly using the zero-shot learning and then using the few-shot learning. We used two different prompts. The first prompt guides a detailed analysis, asking to identify specific text spans for their categories, providing explanations for each of them. It emphasizes a step-by-step approach, where understanding and justification are key. If a category isn't explicitly present, it advises the system to omit it. The first prompt for the zero-shot learning in english was the following one:

```
prompt = ‘‘<s>[INST] You are an expert in detecting elements of the texts. Since
conspiracy narratives are a special type of causal explanation, your task consists in
the recognition of text spans corresponding to the key elements of a text.
Step 1: Identify all of the negative effects mentioned in the text and relate them to
the oppositional narrative. A negative effect is a harmful consequence or negative
impact related to conspiracy theories or critical aspects. Put these negative effects
in the same form that they appear in the text in different lines with the keyword
‘‘NEGATIVE_EFFECT’’.
Step 2: Identify if there is an explicitly stated objective of the oppositional
narrative. An explicit objective refers to a clear and direct statement outlining the
goal or purpose of the narrative being presented. This objective is typically stated
overtly within the text, providing insight into what the proponents of the narrative
are trying to achieve or promote. Put these objectives in the same form that they
appear in the text in different lines with the keyword ‘‘OBJECTIVE’’.
Step 3: Identify if there are victims of the oppositional texts. A victim is a
specific individual or group that is negatively affected by the negative effects
identified in step 1, harmful actions or policies described in the text. Put all
victims with the keyword ‘‘VICTIM’’.
Step 4: Identify if there are conspirators in the text. A conspirator refers to the
entity responsible for planning, executing, or supporting the main action or policy
being discussed in the text. Moreover, a conspirator is responsible for the
NEGATIVE_EFFECTS. Put all the conspirators identified with the keyword ‘‘AGENT’’.
Step 5: Identify if there is any facilitator in the text. A facilitator is a
collaborator or entity that supports the agents in executing the main actions or
```

```
policies discussed in the text. They assist in the achievement of the objectives
outlined by the conspirators, often playing a role in enabling or promoting the
negative effects on the victims. Put all the facilitators identified with the keyword
‘‘FACILITATOR’’.
Step 6: Detect the campaigners that appear in the text. A campaigner is an entity or
someone who unmasks the conspiracy agenda, opposes the conspiracy narrative, and works
to expose or challenge it. Moreover, a campaigner is actively opposing the mainstream
narrative and promoting his own opinion. Put all the campaigners identified with the
keyword ‘‘CAMPAIGNER’’.
Please answer each step with the exact part of the text and explain your answer for
each step. If there is not a specific and clear element, do not provide it.
[/INST]</s> ’’
```

The second prompt is more streamlined, requiring only the span-texts, start and end character positions, and no explanations. It specifies a clear output format in a JSON-like structure, focusing on precision and data extraction. Unlike the first, it simplifies instructions and prioritizes concise, structured output over interpretative details. The second prompt is as follows:

```
prompt = ‘‘<s>[INST] You are an expert in detecting elements of the texts. Since
conspiracy narratives are a special type of causal explanation, your task consists in
the recognition of text spans corresponding to the key elements of a text.
**I want in output only the span-texts with their categories, the start and end char
of the span-text with reference to the input text. You must consider the spaces as
chars as well and start counting from the beginning of the input text. If there is
not a specific and clear element, do not provide it.**
-Identify if there are Agents in the text. They are those responsible for the actions
and/or negative effects described in the comment. In Conspiracy, it could be the
hidden power that pulls the strings. In Critical, it could be the actors that design
the mainstream public health policies. Moreover, a conspirator is responsible for the
NEGATIVE_EFFECTS. Put all the conspirators identified with the keyword ‘‘AGENT’’.
-Identify if there is any facilitator in the text. Facilitators are those who
collaborate with the agents and contribute to the execution of their goals. In
Conspiracy, they could be governments or institutions which, either intentionally or
unwittingly, collaborate with the conspirators and help the conspiracy move forward.
In Critical, they could be healthcare workers, mass media or authority figures who
abide by governmental instructions. Put all the facilitators identified with the
keyword ‘‘FACILITATOR’’.
-Detect the campaigners that appear in the text. Campaigners are those who oppose the
mainstream narrative. In Conspiracy, those who know the truth and expose it to
society at large. In Critical, those who oppose the enforcement of laws and/or refuse
to follow health-related instructions from the authorities. Put all the campaigners
identified with the keyword ‘‘CAMPAIGNER’’.
-Identify if there are victims of the oppositional texts. Victims are those who
```

```
suffer the consequences of the actions and decisions of the agents and/or the
facilitators.  In Conspiracy, the people who are deceived by those in power, and
suffer, become ill, lose their freedom, or die as a result of a hidden plan.  In
Critical, the people who receive the negative consequences of the actions and the
decisions made by those in power, and also suffer, lose their freedom, become ill, or
die as a result of wrong decisions.  Put all victims with the keyword ‘‘VICTIM’’.
-Identify if there is an explicitly stated objective of the oppositional narrative.
Objectives are the intentions and purposes that the agents are trying to achieve.  In
Conspiracy, the goals of the conspirators.  In Critical, the goals of public
authorities, pharmaceutical companies, organizations, etc.  Put these objectives in
the same form that they appear in the text in different lines with the keyword
‘‘OBJECTIVE’’.
-Identify all the negative effects mentioned in the text and relate them to the
oppositional narrative.  They are the negative consequences suffered by the victims as
a result of the actions and decisions of those in power and/or their collaborators.
Put these negative effects in the same form that they appear in the text in different
lines with the keyword ‘‘NEGATIVE_EFFECT’’.
I want an output like this:
"annotations":  [
{
"span_text":
"category":
"start_char":
"end_char":
}
]
[/INST]</s> ’’
```

We also try to see if with the few shot learning the performances would improve. For consistency, we modified the two prompts used for the zero shot learning and added three examples taken from the train dataset. Since *gemma-2-9b-it* was the model that obtained the best results for the zero and few shot learning for the first task, we decided to test it also for the second task. We conducted experiments only for the English dataset. In the following Table 11.9 we can see the results. The second column underlines which prompt was used.

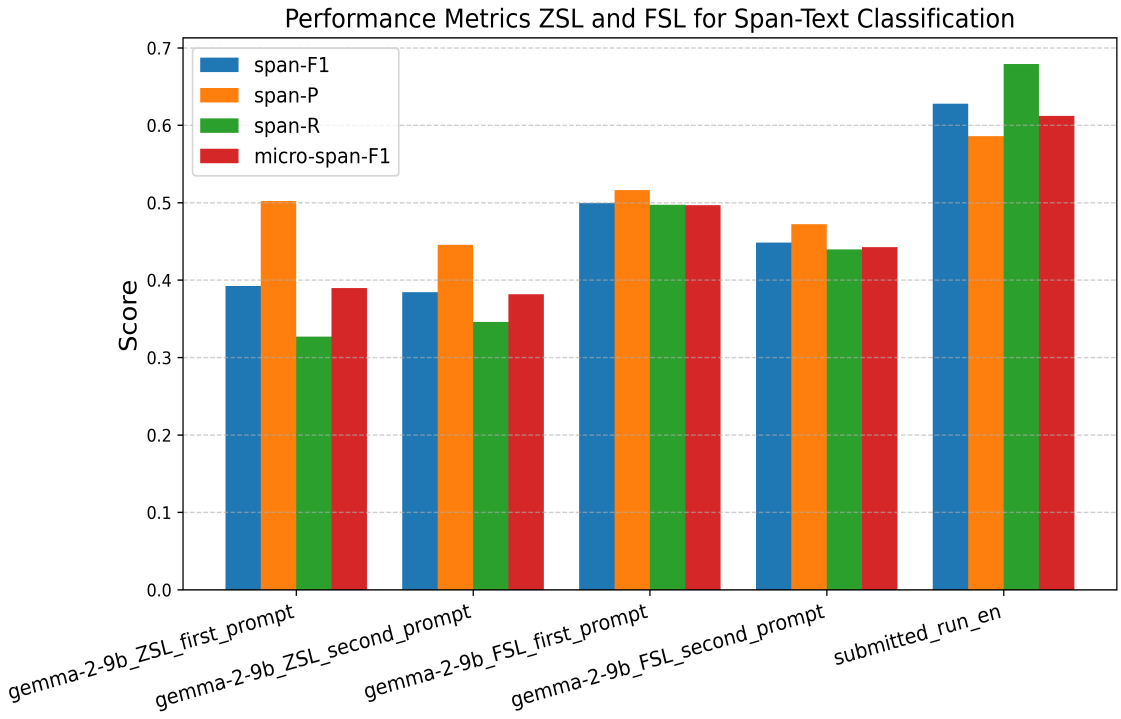


Figure 11.1: Performance metrics of ZSL and FSL for span-text detection

Model	Prompt	span-F1	span-P	span-R	micro-span-F1
ZERO-SHOT					
gemma-2-9b-it	first	**0.3923**	**0.5021**	0.3269	**0.3897**
gemma-2-9b-it	second	0.3843	0.4454	**0.3459**	0.3816
FEW-SHOT					
gemma-2-9b-it	first	**0.4995**	**0.5162**	0.4974	**0.4968**
gemma-2-9b-it	second	0.4485	0.4723	**0.4395**	0.4426
submitted_run		*0.6279*	*0.5859*	*0.6790*	*0.6120*

Table 11.9: Performance metrics of ZSL and FSL for span-text detection

Analyzing Table 11.9, the micro-span-F1 metric clearly illustrates the impact of different learning strategies and prompt designs on the *gemma-2-9b-it* model's performance. In the zero-shot learning setup, the first prompt, which provided detailed explanations, achieved a micro-span-F1 score of 0.3897. Although precision was relatively high at 0.5021, recall was significantly lower at 0.3269, indicating a tendency towards conservative predictions with many missed spans. The simplified second prompt slightly decreased micro-span-

F1 to 0.3816, with precision dropping to 0.4454 but recall improving to 0.3459. This underscores the difficulty in balancing precision and recall without examples to guide the model.

The introduction of few-shot learning led to some improvements. The first FSL prompt increased micro-span-F1 to 0.4968, demonstrating how examples helped the model achieve a better balance between precision (0.5162) and recall (0.4974). Although the second FSL prompt also showed gains over ZSL, its micro-span-F1 was lower at 0.4426, revealing that a simpler prompt still limited the model's overall effectiveness.

Comparing these strategies, FSL had a clear advantage in boosting micro-span-F1, showing the importance of incorporating examples to enhance model confidence and generalization. Nevertheless, the *submitted_run* model outperformed all approaches, achieving a micro-span-F1 of 0.6120, with precision at 0.5859 and recall at 0.6790. This highlights the superiority of fine-tuning, which remains essential for maximizing both precision and recall. These findings emphasize that while prompt design and examples are beneficial, fine-tuning is crucial for achieving the highest performance in complex tasks.

12 | Conclusion and Future Work

In this thesis, we explored the detection of oppositional thinking narratives, specifically focusing on distinguishing between conspiratorial and critical thinking narratives and identifying narrative elements within texts. The research aimed to address the challenges of accurately classifying complex narrative structures using fine-tuned transformer models and LLMs, strategic data processing, data augmentation and ensemble methods. These techniques demonstrated the effectiveness of transformer-based approaches and LLMs approaches in handling nuanced language classification tasks, highlighting the models' adaptability, precision and potential for broader application across languages and domains.

12.1. Concluding Remarks

This research has provided valuable insights by effectively addressing the three research questions and making significant contributions to the detection of oppositional narratives.

First, regarding the ability of transformer models and Large Language Models to distinguish between conspiracy theories and critical thinking narratives, the binary classification experiments confirmed that these models are highly effective. The combination of fine-tuned transformers, Large Language Models, and ensemble methods such as soft voting, led to high MCC scores, demonstrating robust performance in both English and Spanish datasets. This success underscores these models' capability to capture nuanced differences between conspiracy and critical thinking narratives, even when faced with the challenges of imbalanced datasets. The results validate both transformer-based models and LLMs as highly adaptable and precise in managing complex classification tasks across languages and narrative types, effectively distinguishing between these two forms of oppositional thought.

Additionally, experiments with Large Language Models underscored the importance of task-specific fine-tuning. While zero-shot and few-shot learning approaches offered limited effectiveness, fine-tuning allowed the LLMs to achieve state-of-the-art performance,

surpassing the initial systems submitted for the shared task. This outcome highlights the value of model-specific adaptations for optimal results in complex classification tasks. Similarly, experiments with multilingual transformer models confirmed the viability of a unified approach for processing both English and Spanish texts, validating the potential of multilingual binary classification for cross-lingual narrative analysis. These findings reinforce the significance of LLMs and multilingual transformer models in effectively distinguishing and classifying nuanced narratives across different languages and contexts.

Second, in terms of identifying and categorizing narrative elements within texts, this research demonstrated that it is feasible to detect and label key components such as Agents, Facilitators, Victims, Campaigners, Objectives, and Negative Effects within oppositional narratives. The span-level detection task posed notable complexity due to partially overlapping spans and the subtlety of narrative distinctions. However, strategic data preprocessing methods enabled the models to align tokens accurately, critical for precise span detection. Techniques like synonym replacement were essential for improving the models' generalization across linguistic variations without sacrificing precision. The use of multilingual transformer models played a pivotal role in this task, achieving high span-F1 scores across languages, thereby validating the models' capacity for multilingual adaptability and fine-grained identification of narrative components across English and Spanish. This finding reinforces the significance of multilingual transformer models in effective narrative detection across diverse linguistic contexts.

Lastly, concerning the impact of data augmentation techniques on model performance, this thesis demonstrated that these techniques are essential for both binary classification and span-level detection. By addressing class imbalances and promoting generalization across linguistic variations, data augmentation improved MCC scores in binary classification, particularly aiding in the detection of underrepresented conspiracy texts. For span-level detection, augmentation techniques enabled models to handle diverse narrative structures effectively, underscoring their role in refining NLP models to capture the detailed structure of narratives.

12.2. Future Works

Building on the success of the span-level detection task, several promising directions for future research are evident. One key area is the integration of multimodal data, allowing models to process not only text but also images, videos, and audio. Disinformation often relies on non-textual content to amplify its impact, especially on platforms where visual media is prevalent. Extending models to handle multimodal content could enhance

detection efforts, offering a more comprehensive understanding of narrative structures across various media.

Expanding these methods to low-resource languages is another important direction. Although this research achieved strong results in English and Spanish, the techniques can be adapted for languages with fewer resources. Cross-lingual transfer learning, where models trained on high-resource languages are adapted to low-resource ones, could significantly broaden the applicability of these models to address disinformation in regions particularly vulnerable to its effects.

Furthermore, the approaches are adaptable to other languages and domains, which allows for potential real-time applications such as monitoring social media to detect disinformation. However, deploying such systems raises ethical considerations around transparency and accountability to prevent misuse, emphasizing the need for responsible model deployment.

The narrative element detection framework could also benefit from exploring finer distinctions within the taxonomy of narrative elements. Although six fundamental elements were identified, future studies might analyze how narratives evolve over time or adapt to different contexts, enhancing the understanding of disinformation spread and informing more effective countermeasures.

Lastly, future research should focus on improving model interpretability. Although the models in this thesis performed well, their decision-making processes remain opaque. Techniques like LIME [85] or SHAP [60] could provide clearer insights into model predictions, an essential step for applications in high-stakes contexts such as content moderation or legal frameworks. Enhancing transparency will be key to ensuring these models are used ethically, responsibly, and effectively in broader applications.

Bibliography

[1] L. B. Y. Ai, C. Ai, and R. Ai. Gpt-4 technical report.

[2] H. Alibrahim and S. A. Ludwig. Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 1551–1559. IEEE, 2021.

[3] H. Allcott and M. Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–236, 2017. doi: 10.1257/jep.31.2.211.

[4] D. Altinok. *Mastering spaCy: An end-to-end practical guide to implementing NLP applications using the Python ecosystem*. Packt Publishing Ltd, 2021.

[5] M. Azimi and G. Pekcan. Structural health monitoring using extremely-compressed data through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, 06 2020. doi: 10.1111/mice.12517.

[6] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 2015 International Conference on Learning Representations*, 2014.

[7] E. Bassignana, V. Basile, V. Patti, et al. Hurtlex: A multilingual lexicon of words to hurt. In *CEUR Workshop proceedings*, volume 2253, pages 1–6. CEUR-WS, 2018.

[8] I. Beltagy, K. Lo, and A. Cohan. SciBERT: A pretrained language model for scientific text. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1371.

[9] J. Bevendorff, X. Bonet-Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, and P. Rosso. Overview of PAN 2024. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2024)*, Lecture Notes in Computer Science, Berlin Heidelberg New York, Sept. 2024. Springer.

[10] A. Bovet and H. A. Makse. Influence of fake news in twitter during the 2016 us presidential election. *Nature communications*, 10(1):7, 2019.

[11] A. Brandolini. Bullshit asymmetry principle. 2013. URL `https://en.wikipedia.org/wiki/Brandolini%27s_law`. Accessed: 2024-09-15.

[12] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[13] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[14] J. S. Brennen, F. M. Simon, P. N. Howard, and R. K. Nielsen. Types, sources, and claims of covid-19 misinformation. 2020.

[15] J. D. Brown. Point-biserial correlation coefficients. *Statistics*, 5(3):12–6, 2001.

[16] L. Bursztyn, A. Rao, C. P. Roth, and D. H. Yanagizawa-Drott. Misinformation during a pandemic. Technical report, National Bureau of Economic Research, 2020.

[17] C. P. Carrino, J. Llop, M. Pàmies, A. Gutiérrez-Fandiño, J. Armengol-Estapé, J. Silveira-Ocampo, A. Valencia, A. Gonzalez-Agirre, and M. Villegas. Pretrained biomedical language models for clinical nlp in spanish. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 193–199, 2022.

[18] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez. Spanish pre-trained BERT model and evaluation data. In *Practical ML for Developing Countries Workshop at the International Conference on Learning Representations (ICLR 2020)*, 2020.

[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[20] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[21] D. Chicco and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020.

[22] M. Cinelli, G. De Francisci Morales, A. Galeazzi, W. Quattrociocchi, and M. Starnini. The echo chamber effect on social media. *Proceedings of the National Academy of Sciences*, 118(9): e2023301118, 2021.

[23] A. Conneau. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[24] G. Da San Martino, A. Barrón-Cedeño, H. Wachsmuth, R. Petrov, and P. Nakov. Semeval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, 2020.

[25] R. Dale, H. Moisl, and H. Somers. *Handbook of natural language processing*. CRC press, 2000.

[26] J. de la Rosa, E. G. Ponferrada, M. Romero, P. Villegas, P. González de Prado Salas, and M. Grandury. Bertin: Efficient pre-training of a spanish language model using perplexity sampling. *Procesamiento del Lenguaje Natural*, 68(0):13–23, 2022. ISSN 1989-7553.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

[28] K. M. Douglas and R. M. Sutton. What are conspiracy theories? a definitional approach to their correlates, consequences, and communication. *Annual review of psychology*, 74:271–298, 2023.

[29] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[30] J. Ekström. The phi-coefficient, the tetrachoric correlation coefficient, and the pearson-yule debate. 2011.

[31] Encord. Classification metrics: Accuracy, precision, and recall, 2023. URL `https://encord.com/blog/classification-metrics-accuracy-precision-recall/`. Accessed: 2024-11-06.

[32] E. Ferrara. Disinformation and social bot operations in the run up to the 2017 french presidential election. *arXiv preprint arXiv:1707.00086*, 2017.

[33] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[34] A. Giachanou, B. Ghanem, and P. Rosso. Detection of conspiracy propagators using psycho-linguistic characteristics. *Journal of Information Science*, 49(1):3–17, 2023.

[35] T. Gillespie. *Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media*. Yale University Press, 2018.

[36] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[37] L. Graves. *Deciding what's true: The rise of political fact-checking in American journalism*. Columbia University Press, 2016. ISBN 9780231175066.

[38] I. Grisanti. Named entity recognition with llms: Extract conversation metadata, 2024. URL `https://medium.com/@grisanti.isidoro/named-entity-recognition-with-llms-extract-conversation-metadata-94d5536178f2`. Accessed: 2024-11-11.

[39] N. J. Guess A. and T. J. Less than you think: Prevalence and predictors of fake news dissemination on facebook. *Science Advances*, 5(1):eaau4586, 2019. doi: 10.1126/sciadv.aau4586.

[40] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, C. Armentano-Oller, C. Rodriguez-Penagos, A. Gonzalez-Agirre, and M. Villegas. Maria: Spanish language models. *Procesamiento del Lenguaje Natural*, 68:39–60, 2022.

[41] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[42] P. Holur, T.-Y. Wang, S. Shahsavari, T. R. Tangherlini, and V. P. Roychowdhury. Which side are you on? insider-outsider classification in conspiracy-theoretic social media. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4975–4987. ACL, 2022.

[43] E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. Lora: Low-rank adaptation of large language models.

[44] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning*, pages 137–142. Springer, 1998.

[45] K. S. Jones and E. Barber. What makes an automatic keyword classification effective? *Journal of the American Society for Information Science*, 22(3):166–175, 1971.

[46] D. Jurafsky and J. H. Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

[47] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154, 2017.

[48] D. Korenčić, B. Chulvi, X. B. Casals, M. Taulé, P. Rosso, and F. Rangel. Overview of the oppositional thinking analysis pan task at clef 2024. In *Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum*, 2024.

[49] D. Korenčić, B. Chulvi, X. B. Casals, A. Toselli, M. Taulé, and P. Rosso. What distinguishes conspiracy from critical narratives? a computational analysis of oppositional discourse. *Expert Systems*, page e13671, 2024. doi: https://doi.org/10.1111/exsy.13671. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.13671`.

[50] L. I. Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

[51] J. Lafferty, A. McCallum, F. Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml*, volume 1, page 3. Williamstown, MA, 2001.

[52] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. In K. Knight, A. Nenkova, and O. Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1030. URL `https://aclanthology.org/N16-1030`.

[53] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, 2016.

[54] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, et al. The science of fake news. *Science*, 359(6380): 1094–1096, 2018.

[55] S. Lewandowsky, U. K. Ecker, and J. Cook. Beyond misinformation: Understanding and coping with the "post-truth" era. *Journal of applied research in memory and cognition*, 6(4):353–369, 2017.

[56] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(2):1–41, 2022.

[57] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A

systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[58] Y. Liu and M. Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, 2019.

[59] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. URL `https://api.semanticscholar.org/CorpusID:198953378`.

[60] S. Lundberg. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[61] A. Manconi, G. Armano, M. Gnocchi, and L. Milanesi. A soft-voting ensemble classifier for detecting patients affected by covid-19. *Applied Sciences*, 12(15):7554, 2022. doi: 10.3390/app12157554.

[62] W. Marcellino, T. C. Helmus, J. Kerrigan, H. Reininger, R. I. Karimov, and R. A. Lawrence. Detecting conspirac y theories on social media, 2021.

[63] A. E. Marwick and R. Lewis. Media manipulation and disinformation online. *Data & Society Research Institute*, 2017.

[64] S. McGrew, T. Ortega, J. Breakstone, and S. Wineburg. The challenge that's bigger than fake news: Civic reasoning in a social media environment. *American educator*, 41(3):4, 2017.

[65] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[66] T. M. Mitchell and T. M. Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[67] R. S. Mueller III. Report on the investigation into russian interference in the 2016 presidential election, volumes i and ii (redacted version of april 18, 2019). 2019.

[68] MyScale Team. Unlocking llama 3.1: Comprehensive insights & benchmarks, 2024. URL `https://myscale.com/blog/top-5-insights-llama-3-1-performance-benchmarks/`.

[69] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[70] J. Ni, G. H. Abrego, N. Constant, J. Ma, K. Hall, D. Cer, and Y. Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, 2022.

[71] R. Odegua. An empirical study of ensemble techniques (bagging, boosting and stacking). In *Proc. Conf.: Deep Learn. IndabaXAt*, 2019.

[72] W. H. Organization. Managing the covid-19 infodemic: Promoting healthy behaviors and mitigating the harm from misinformation and disinformation. 2020.

[73] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and

E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[74] G. Pennycook and D. G. Rand. Fighting misinformation on social media using crowdsourced judgments of news source quality. *Proceedings of the National Academy of Sciences of the United States of America*, 116(7):2521, 2019.

[75] Y. Peskine, D. Korenčić, I. Grubisic, P. Papotti, R. Troncy, and P. Rosso. Definitions matter: Guiding gpt for multi-label classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4054–4063, 2023.

[76] T. Pires, E. Schlinger, and D. Garrette. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, 2019.

[77] K. Pogorelov, J. Langguth, and P. Halvorsen. Fakenews: Corona virus and 5g conspiracy multimedia analysis task at mediaeval 2020. In *Proceedings of the MediaEval 2020 Multimedia Evaluation Workshop*. CEUR-WS, 2020.

[78] K. Pogorelov, J. Langguth, and P. Halvorsen. Fakenews: Corona virus and conspiracies multimedia analysis task at mediaeval 2021. In *Proceedings of the MediaEval 2021 Multimedia Evaluation Workshop*. CEUR-WS, 2021.

[79] K. Pogorelov, J. Langguth, and P. Halvorsen. Fakenews detection task at mediaeval 2022: Multimodal approaches to conspiracies. In *Proceedings of the MediaEval 2022 Multimedia Evaluation Workshop*. CEUR-WS, 2022.

[80] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso. An evaluation framework for plagiarism detection. In C.-R. Huang and D. Jurafsky, editors, *Coling 2010: Posters*, pages 997–1005, Beijing, China, Aug. 2010. Coling 2010 Organizing Committee. URL `https://aclanthology.org/C10-2115`.

[81] D. M. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *International Journal of Machine Learning Technology*, 2(1):37–63, 2020.

[82] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286. Ieee, 1989.

[83] F. Rangel, A. Giachanou, B. H. H. Ghanem, and P. Rosso. Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter. In *CEUR workshop proceedings*, volume 2696, pages 1–18. Sun SITE Central Europe, 2020.

[84] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Conference on empirical methods in natural language processing*, 1996.

[85] M. T. Ribeiro, S. Singh, and C. Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[86] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[87] E. T. K. Sang and S. Buchholz. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop (CONLL/LLL 2000). Lissabon, Portugal, 13-14 september 2000*, pages 127–132. ACL, 2000.

[88] E. T. K. Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[89] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34 (1):1–47, 2002.

[90] C. Shao, G. L. Ciampaglia, O. Varol, A. Flammini, and F. Menczer. The spread of fake news by social bots. *arXiv preprint arXiv:1707.07592*, 96(104):14, 2017.

[91] A. Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[92] L. Souder. The ethics of scholarly peer review: a review of the literature. *Learned Publishing*, 24 (1):55–72, 2011.

[93] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[94] V. Teller. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2000.

[95] J. Tiedemann and S. Thottingal. Opus-mt–building open translation services for the world. In *Proceedings of the 22nd annual conference of the European Association for Machine Translation*, pages 479–480, 2020.

[96] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 human language technology conference of the north american chapter of the association for computational linguistics*, pages 252–259, 2003.

[97] A. Tulbure and M. Coll Ardanuy. Conspiracy vs critical thinking using an ensemble of transformers with data augmentation techniques. *Working Notes of CLEF*, 2024.

[98] UNESCO. The covid-19 conspiracy theory: Online data analysis and its impact. `https://www.unesco.org/en/articles/fightbiodiversityloss/covid-19`, 2020. Accessed: 2024-09-15.

[99] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[100] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *science*, 359(6380): 1146–1151, 2018.

[101] I. Vykopal, M. Pikuliak, I. Srba, R. Moro, D. Macko, and M. Bielikova. Disinformation capabilities of large language models. *arXiv preprint arXiv:2311.08838*, 2023.

[102] A. Wang. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[103] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Transformers: State-of-the-art natural language processing. *EMNLP 2020*, page 38, 2020.

[104] D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[105] F. Xiong, T. Markchom, Z. Zheng, S. Jung, V. Ojha, and H. Liang. Fine-tuning large language models for multigenerator, multidomain, and multilingual machine-generated text detection. *arXiv preprint arXiv:2401.12326*, 2024.

[106] V. Yadav and S. Bethard. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, 2018.

[107] G. Yenduri, M. Ramalingam, G. C. Selvi, Y. Supriya, G. Srivastava, P. K. R. Maddikunta, G. D. Raj, R. H. Jhaveri, B. Prabadevi, W. Wang, et al. Gpt (generative pre-trained transformer)–a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*, 2024.

[108] Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.

A | Appendix A

In this appendix, we provide extended data analysis focused on the relationships and patterns between different annotation categories in the datasets.

A.1. Data Analysis Extended

Figure A.1 and Figure A.2 show correlation between the occurrence of at least one span element with other span elements for Spanish and for both datasets, respectively.

Figure A.3 shows the pairwise correlation between different annotation categories for a given dataset. Each bar represents the strength of the correlation between two annotation categories, indicating how often they appear together across the dataset. The higher the bar, the stronger the correlation between that pair of categories. The correlation values can range from -1 to 1, where: 1 means a perfect positive correlation (the two categories always appear together), 0 means no correlation (the presence of one category doesn't affect the other) and -1 means a perfect negative correlation (the two categories never appear together).

In the English dataset, the strongest correlation is between *Negative_Effect* and *Victim* (0.28), suggesting that texts labeled with Victim often include a *Negative_Effect*, which makes intuitive sense, while *Agent* and *Negave_Effect* have a weak negative correlation (-0.08), indicating that these categories rarely co-occur.

For the Spanish dataset the strongest correlation is between *Victim* and *Negative_Effect* (0.30), again aligning with the logical connection that a *Victim* tends to experience a *Negative_Effect*. A negative correlation between *Objective* and *Negative_Effect* (-0.16) is more pronounced in the Spanish dataset than in English, suggesting a clearer division between these categories.

Figure A.4 is shown the mean number of times an annotation shows up for each category label in both datasets.

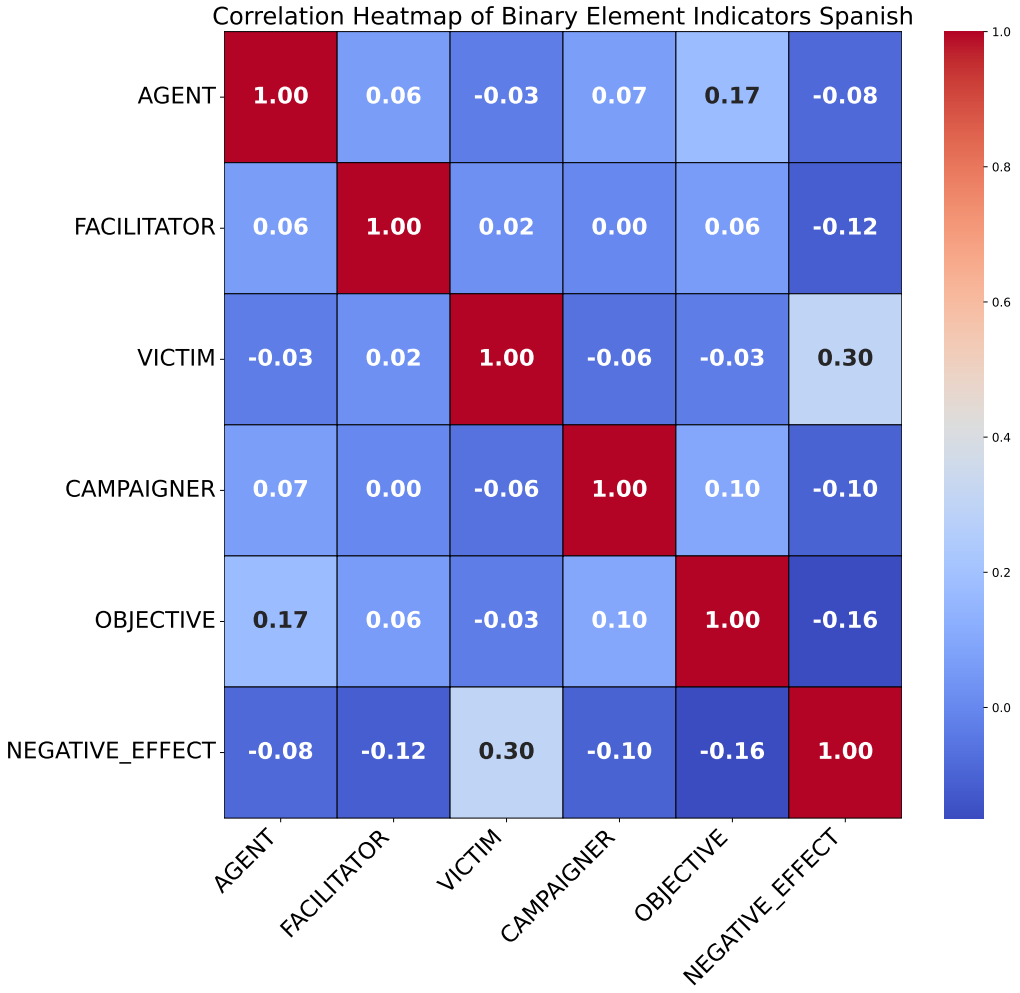


Figure A.1: Correlation between the presence of at least one span element and other span elements in the Spanish dataset

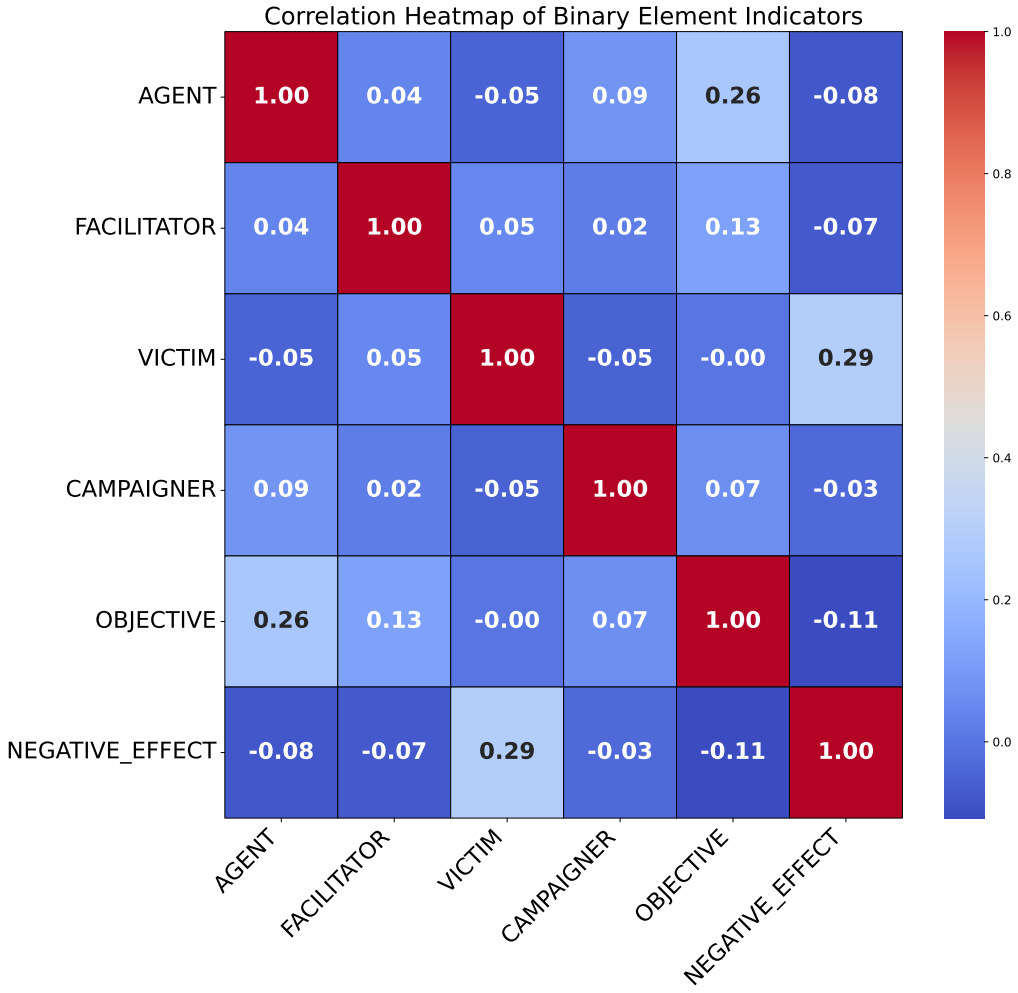


Figure A.2: Correlation between the presence of at least one span element and other span elements in both datasets

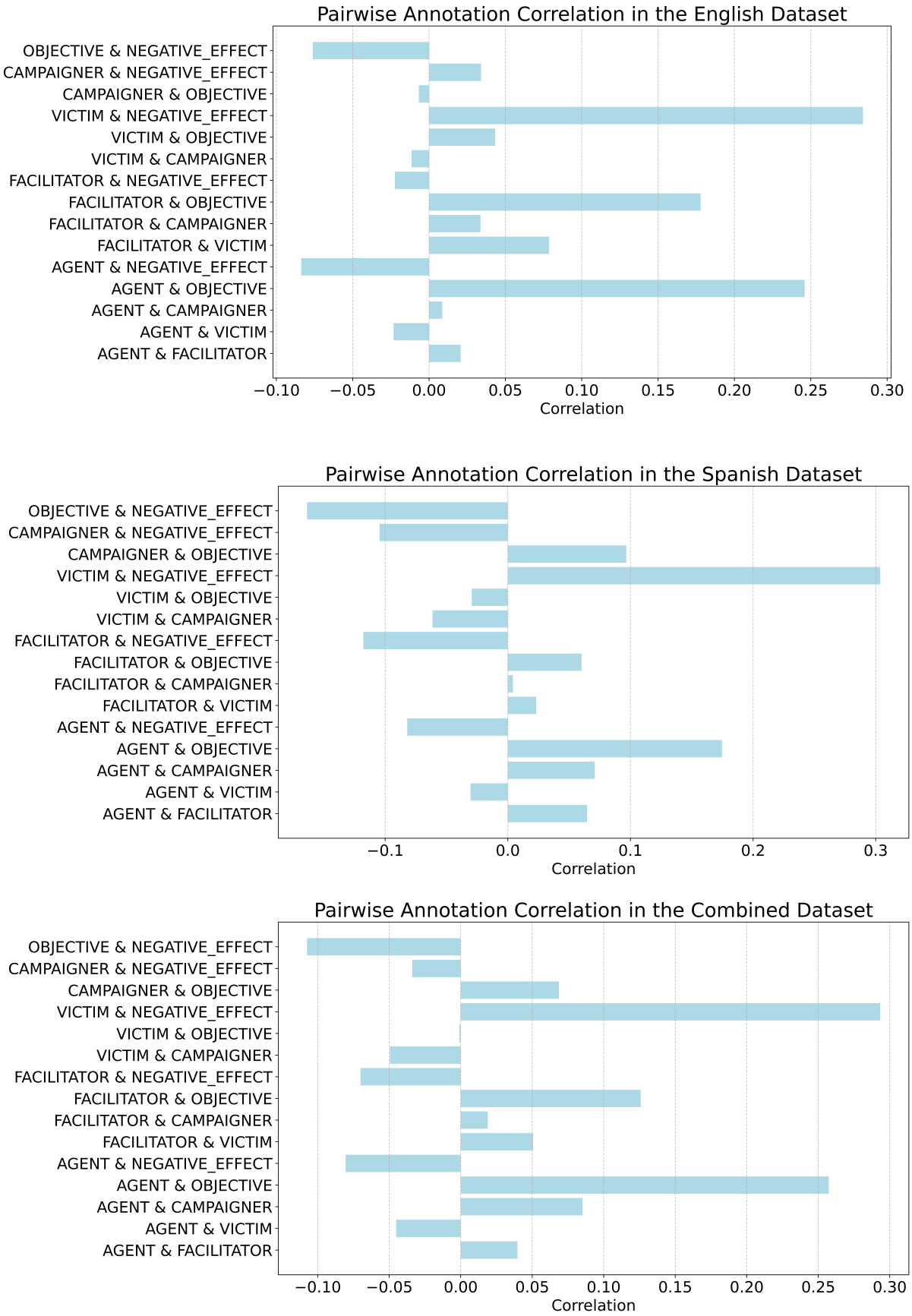
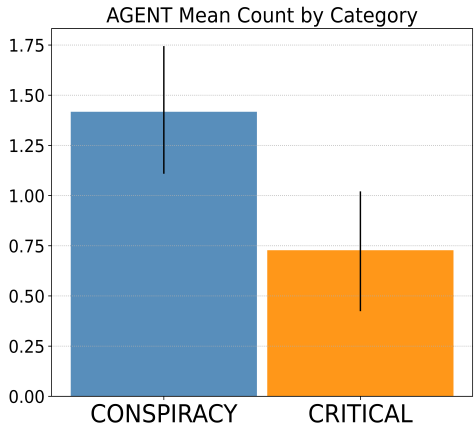
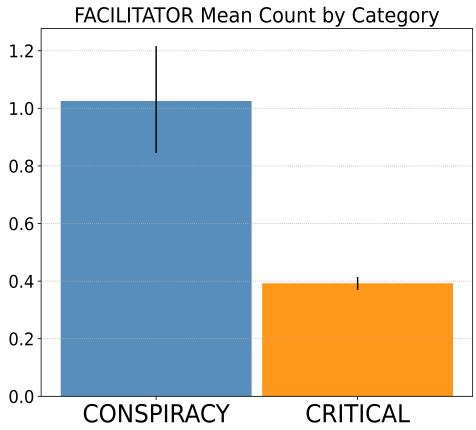


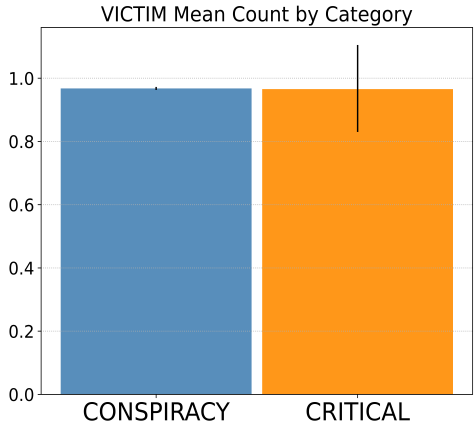
Figure A.3: Pairwise annotation correlations across datasets



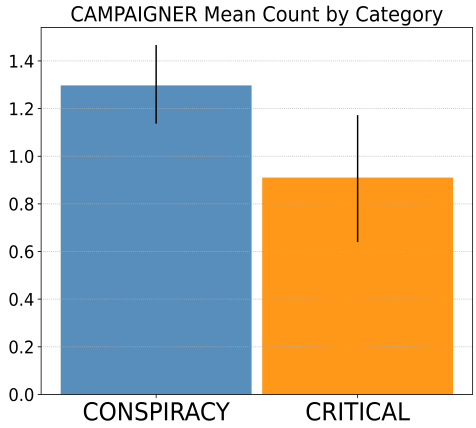
(a) Mean annotations for AGENT.



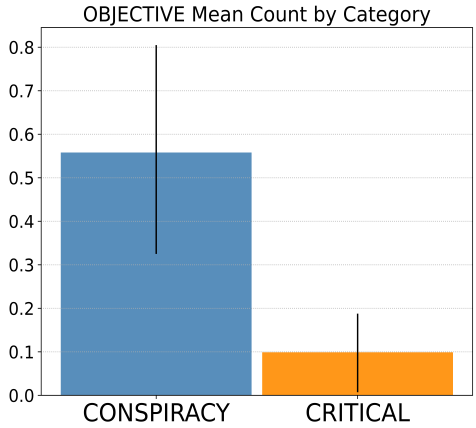
(b) Mean annotations for FACILITATOR.



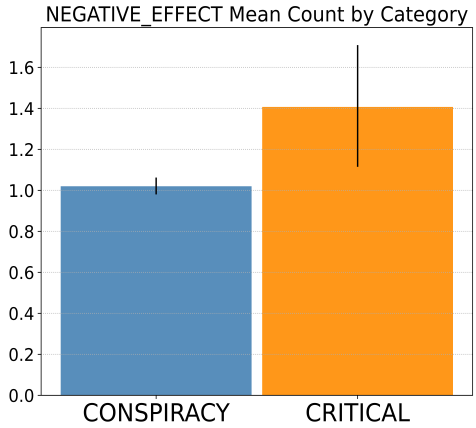
(c) Mean annotations for VICTIM.



(d) Mean annotations for CAMPAIGNER.



(e) Mean annotations for OBJECTIVE.



(f) Mean annotations for NEGATIVE EFFECT.

Figure A.4: Mean number of times an annotation shows up for each category label in both datasets.

B | Appendix B

In this appendix, we provide the official rankings and performance metrics for Tasks 1 and 2 in both English and Spanish [48].

B.1. Detailed Results for Task 1 in English

In the following table (Table B.1) are detailed the results and rankings of the teams participating on task 1, binary classification of text as either conspiracy or critical, for English texts. Performance metrics are: Matthews Correlation Coefficient, macro-averaged F1, and per-class binary F1's.

Pos.	Team	MCC	F1-Macro	F1-Conspiracy	F1-Critical
1	IUCL	0.8388	0.9194	0.8947	0.9441
2	AI_Fusion	0.8303	0.9147	0.8866	0.9429
3	SINAI	0.8297	0.9149	0.8886	0.9412
4	ezio	0.8212	0.9097	0.8792	0.9402
5	hinlole	0.8198	0.9098	0.8811	0.9386
6	Zleon	0.8195	0.9096	0.8804	0.9388
7	virmel	0.8192	0.9092	0.8793	0.9391
8	inaki	0.8149	0.9072	0.8770	0.9374
9	yeste	0.8124	0.9057	0.8746	0.9368
10	auxR	0.8088	0.9043	0.8739	0.9347
11	Elias&Sergio	0.8034	0.9012	0.8687	0.9338
12	theateam	0.8031	0.8999	0.8650	0.9347
13	trustno1	0.7983	0.8991	0.8675	0.9307
14	DSVS	0.7970	0.8985	0.8674	0.9296
15	sail	0.7969	0.8978	0.8687	0.9268
16	ojo-bes	0.7969	0.8981	0.8648	0.9314
17	RD-IA-FUN	0.7965	0.8977	0.8636	0.9317
	baseline-BERT	0.7964	0.8975	0.8632	0.9318
18	aish_team	0.7917	0.8944	0.8580	0.9309
19	rfenthusiasts	0.7902	0.8948	0.8605	0.9291
20	Dap_upv	0.7898	0.8944	0.8593	0.9294
21	oppositional_opposition	0.7894	0.8935	0.8571	0.9300
22	miqarn	0.7881	0.8938	0.8593	0.9283

Table B.1: Results for task 1 - English [48]

Pos.	Team	MCC	F1-Macro	F1-Conspiracy	F1-Critical
23	CHEEXIST	0.7875	0.8932	0.8576	0.9287
24	tulbure	0.7872	0.8917	0.8536	0.9297
25	XplaiNLP	0.7871	0.8922	0.8550	0.9294
26	TheGymNerds	0.7854	0.8923	0.8567	0.9278
27	nlpln	0.7844	0.8922	0.8580	0.9263
28	RalloRico	0.7771	0.8879	0.8559	0.9198
29	LasGarcias	0.7758	0.8855	0.8447	0.9263
30	zhengqiaozeng	0.7758	0.8866	0.8476	0.9256
31	ALC-UPV-JD-2	0.7725	0.8860	0.8491	0.9230
32	LorenaEloy	0.7713	0.8847	0.8455	0.9239
33	lnr-alhu	0.7708	0.8853	0.8488	0.9219
34	NACKO	0.7692	0.8838	0.8446	0.9230
35	paranoia-pulverizers	0.7680	0.8838	0.8462	0.9215
36	DiTana	0.7653	0.8806	0.8490	0.9123
37	FredYNed	0.7643	0.8806	0.8392	0.9220
38	dannuchihaxxx	0.7643	0.8801	0.8377	0.9224
39	lnrdetectives	0.7631	0.8806	0.8472	0.9141
40	TargaMarhuenda	0.7617	0.8807	0.8424	0.9190
41	Trainers	0.7596	0.8797	0.8412	0.9182
42	thetaylorswiftteam	0.7577	0.8755	0.8302	0.9208
43	locasporlnr	0.7575	0.8787	0.8399	0.9174
44	lnradri	0.7552	0.8759	0.8326	0.9192
45	TokoAI	0.7542	0.8767	0.8363	0.9172
46	ede	0.7539	0.8769	0.8384	0.9155
47	lnrverdnav	0.7529	0.8746	0.8308	0.9185
48	lnrdahe	0.7488	0.8736	0.8308	0.9163
49	epistemologos	0.7486	0.8742	0.8341	0.9143
50	lucia&ainhoa	0.7473	0.8733	0.8316	0.9150
51	pistacchio	0.7414	0.8678	0.8200	0.9155
52	lnrBraulioPaula	0.7393	0.8658	0.8165	0.9152
53	Marc_Coral	0.7392	0.8663	0.8176	0.9150
54	Ramon&Cajal	0.7284	0.8633	0.8169	0.9096
55	lnr-lladrogal	0.7253	0.8603	0.8106	0.9100
56	lnr-fanny-nuria	0.7253	0.8594	0.8082	0.9106
57	MarcosJavi	0.7190	0.8583	0.8097	0.9069
58	lnr-cla	0.7168	0.8573	0.8085	0.9061
59	lnr-jacobantonio	0.7168	0.8573	0.8085	0.9061
60	MUCS	0.7162	0.8538	0.7994	0.9082
61	lnr-aina-julia	0.7157	0.8574	0.8102	0.9046
62	LaDolceVita	0.7072	0.8519	0.8000	0.9037
63	alopfer	0.7056	0.8518	0.8012	0.9023
64	lnr-luqrud	0.7056	0.8518	0.8012	0.9023
65	LNR-JoanPau	0.7051	0.8426	0.7793	0.9058
66	lnr-carla	0.7000	0.8476	0.7932	0.9020
67	lnr-Inetum	0.6981	0.8328	0.7617	0.9039
68	lnr-antonio	0.6852	0.8300	0.7598	0.9002
69	LluisJorge	0.6784	0.8382	0.7830	0.8934
70	anselmo-team	0.6725	0.8341	0.7752	0.8930
71	lnr-pavid	0.5959	0.7974	0.7297	0.8651
72	LNRMADME	0.5469	0.7717	0.6914	0.8521

Table B.1: Results for task 1 - English (cont.)[48]

Pos.	Team	MCC	F1-Macro	F1-Conspiracy	F1-Critical
73	lnr-mariagb_elenaog	0.5069	0.7250	0.5966	0.8534
74	LNR_08	0.4429	0.6834	0.5276	0.8391
75	Kaprov	0.3700	0.6240	0.4224	0.8255
76	lnr_cebusqui	0.0482	0.4760	0.1847	0.7674
77	jtommor	0.0403	0.5167	0.3312	0.7023
78	eledu	-0.4598	0.2350	0.2740	0.1960
79	david-canet	-0.6310	0.1632	0.1883	0.1381
80	lnr-guilty	-0.6595	0.1433	0.2247	0.0619
81	lnrANRI	-0.7551	0.1072	0.1474	0.0670
82	ROCurve	-0.8009	0.0884	0.1112	0.0656

Table B.1: Results for task 1 - English (cont.)[48]

B.2. Detailed Results for Task 1 in Spanish

In the following table (Table B.2) are detailed the results and rankings of the teams participating on task 1, binary classification of text as either conspiracy or critical, for Spanish texts. Performance metrics are: Matthews Correlation Coefficient, macro-averaged F1, and per-class binary F1's.

Pos.	Team	MCC	F1-Macro	F1-Conspiracy	F1-Critical
1	SINAI	0.7429	0.8705	0.8319	0.9091
2	auxR	0.7205	0.8572	0.8112	0.9032
3	RD-IA-FUN	0.7028	0.8497	0.8035	0.8960
4	Elias&Sergio	0.6971	0.8485	0.8087	0.8884
5	AI_Fusion	0.6872	0.8419	0.7931	0.8908
6	zhengqiaozeng	0.6871	0.8417	0.7925	0.8909
7	virmel	0.6854	0.8426	0.8022	0.8831
8	trustno1	0.6848	0.8400	0.7895	0.8906
9	Zleon	0.6826	0.8410	0.7955	0.8865
10	ojo-bes	0.6817	0.8395	0.8026	0.8764
11	tulbure	0.6722	0.8293	0.7699	0.8887
12	sail	0.6719	0.8299	0.7713	0.8884
13	nlpln	0.6681	0.8339	0.7872	0.8806
	baseline-BETO	0.6681	0.8339	0.7872	0.8806
14	pistacchio	0.6678	0.8327	0.7822	0.8833
15	rfenthusiasts	0.6656	0.8255	0.7643	0.8868
16	XplaiNLP	0.6622	0.8274	0.7708	0.8840
17	yeste	0.6609	0.8291	0.7770	0.8812
18	oppositional_opposition	0.6601	0.8274	0.7724	0.8825
19	epistemologos	0.6562	0.8264	0.7728	0.8801
20	miqarn	0.6562	0.8264	0.7728	0.8801

Table B.2: Results for task 1 - Spanish [48]

Pos.	Team	MCC	F1-Macro	F1-Conspiracy	F1-Critical
21	theateam	0.6557	0.8252	0.7695	0.8810
22	ezio	0.6535	0.8242	0.7683	0.8801
23	lucia&ainhoa	0.6524	0.8260	0.7765	0.8754
24	TargaMarhuenda	0.6516	0.8240	0.7692	0.8787
25	TokoAI	0.6516	0.8240	0.7692	0.8787
26	paranoia-pulverizers	0.6494	0.8246	0.7762	0.8730
27	NACKO	0.6467	0.8232	0.7739	0.8726
28	ALC-UPV-JD-2	0.6467	0.8227	0.7705	0.8748
29	DSVS	0.6462	0.8231	0.7753	0.8709
30	RD-IA-FUN	0.6445	0.8160	0.7523	0.8796
31	locasporlnr	0.6437	0.8216	0.7709	0.8723
32	DiTana	0.6377	0.8187	0.7677	0.8696
33	lnr-BraulioPaula	0.6358	0.8173	0.7731	0.8615
34	Dap_upv	0.6306	0.8115	0.7493	0.8737
35	TheGymNerds	0.6306	0.8106	0.7470	0.8743
36	MUCS	0.6293	0.8060	0.7363	0.8756
37	LasGarcias	0.6247	0.8122	0.7594	0.8649
38	lnr-dahe	0.6196	0.8066	0.7437	0.8694
39	lnr-adri	0.6194	0.8060	0.7422	0.8698
40	hinlole	0.6192	0.8048	0.7391	0.8706
41	RalloRico	0.6105	0.8018	0.7370	0.8666
42	lnr-aina-julia	0.6103	0.7978	0.7264	0.8692
43	lnr-verdnav	0.6101	0.7991	0.7298	0.8684
44	thetaylorswiftteam	0.6066	0.8025	0.7436	0.8613
45	lnr-alhu	0.6024	0.7991	0.7358	0.8624
46	lnr-luqrud	0.6010	0.7945	0.7237	0.8654
47	lnr-lladrogal	0.5967	0.7942	0.7256	0.8627
48	ede	0.5965	0.7967	0.7341	0.8593
49	Fred&Ned	0.5931	0.7940	0.7283	0.8597
50	LaDolceVita	0.5921	0.7818	0.6981	0.8656
51	LNR-JoanPau	0.5920	0.7916	0.7218	0.8614
52	anselmo-team	0.5899	0.7860	0.7085	0.8634
53	Ramon&Cajal	0.5858	0.7916	0.7281	0.8552
54	lnr-fanny-nuria	0.5813	0.7874	0.7181	0.8567
55	lnr-antonio	0.5736	0.7816	0.7071	0.8561
56	LluisJorge	0.5690	0.7750	0.6929	0.8571
57	lnr-cla	0.5651	0.7788	0.7055	0.8520
58	lnr-jacobantonio	0.5651	0.7788	0.7055	0.8520
59	lnr-pavid	0.5569	0.7771	0.7089	0.8453
60	alopfer	0.5520	0.7727	0.6984	0.8470
61	LNRMADME	0.5490	0.7704	0.6937	0.8471
62	lnr-carla	0.5484	0.7686	0.6890	0.8482
63	LorenaEloy	0.5433	0.7621	0.6751	0.8492
64	CHEEXIST	0.5379	0.5995	0.5621	0.5456
65	lnr-guilty	0.5273	0.7620	0.6880	0.8360
66	eledu	0.5057	0.7263	0.6098	0.8429
67	lnr-mariagb_elenaog	0.4966	0.7325	0.6271	0.8379
68	dannuchihaxxx	0.4727	0.7310	0.6382	0.8238
69	lnr-detectives	0.4029	0.6734	0.6509	0.6960
70	LNR_08	0.0608	0.4771	0.2000	0.7542

Table B.2: Results for task 1 - Spanish (cont.) [48]

Pos.	Team	MCC	F1-Macro	F1-Conspiracy	F1-Critical
71	jtommor	0.0105	0.5051	0.3813	0.6288
72	lnr-Inetum	0.0000	0.3880	0.0000	0.7760
73	Marc_Coral	0.0000	0.2679	0.5359	0.0000
74	MarcosJavi	-0.0389	0.3887	0.0054	0.7720
75	lnr_cebusqui	-0.4112	0.2481	0.3466	0.1496
76	david-canet	-0.5058	0.2114	0.3029	0.1199
77	lnrANRI	-0.6146	0.1766	0.1939	0.1593
78	ROCurve	-0.6457	0.1628	0.1770	0.1485

Table B.2: Results for task 1 - Spanish (cont.) [48]

B.3. Detailed Results for Task 2 in English

In the following table (Table B.3) are detailed the results and rankings of the teams participating on task 2, token classification of span-level narrative elements, for English texts. The performance metrics are: span-F1 (macro-averaged over span labels), span-precision, span-recall, and micro-averaged span-F1.

Position	Team	span-F1	span-P	span-R	micro-span-F1
1	tulbure	0.6279	0.5859	0.6790	0.6120
2	Zleon	0.6089	0.5537	0.6881	0.5856
3	hinlole	0.5886	0.5243	0.6834	0.5571
4	oppositional_opposition	0.5866	0.5347	0.6586	0.5344
5	AI_Fusion	0.5805	0.5585	0.6082	0.5437
6	virmel	0.5742	0.5235	0.6477	0.5540
7	miqarn	0.5739	0.5184	0.6462	0.5325
8	TargaMarhuenda	0.5701	0.5161	0.6477	0.5437
9	ezio	0.5694	0.5229	0.6340	0.5389
10	zhengqiaozeng	0.5666	0.5122	0.6485	0.5421
11	Elias&Sergio	0.5627	0.5149	0.6364	0.5248
12	DSVS	0.5598	0.5332	0.6012	0.5287
13	CHEEXIST	0.5524	0.4767	0.6845	0.5299
14	rfenthusiasts	0.5479	0.5381	0.5666	0.5408
15	ALC-UPV-JD-2	0.5377	0.4643	0.6562	0.4956
	baseline-BERT	0.5323	0.4684	0.6334	0.4998
16	Dap_upv	0.5272	0.4617	0.6297	0.4973
17	aish_team	0.5213	0.4181	0.7456	0.2571
18	SINAI	0.4582	0.5553	0.4279	0.4571
19	Trainers	0.3382	0.5124	0.2609	0.2858
20	nlpln	0.3339	0.5286	0.3303	0.2710
21	ROCurve	0.2996	0.3154	0.3031	0.3425
22	TokoAI	0.2760	0.1870	0.6119	0.2677
23	DiTana	0.2756	0.5259	0.1947	0.2599

Table B.3: Results for task 2 - English [48]

Position	Team	span-F1	span-P	span-R	micro-span-F1
24	TheGymNerds	0.2070	0.2076	0.2127	0.2329
25	epistemologos	0.1709	0.1286	0.3244	0.1201
25	epistemologos	0.1709	0.1286	0.3244	0.1201
26	theateam	0.1503	0.1401	0.1652	0.0387
27	LaDolceVita	0.0726	0.2040	0.0453	0.0630
28	kaprov	0.0150	0.0261	0.0165	0.0600

Table B.3: Results for task 2 - English (cont.) [48]

B.4. Detailed Results for Task 2 in Spanish

In the following table (Table B.4) are detailed the results and rankings of the teams participating on task 2, token classification of span-level narrative elements, for Spanish texts. The performance metrics are: span-F1 (macro-averaged over span labels), span-precision, span-recall, and micro-averaged span-F1.

Position	Team	span-F1	span-P	span-R	micro-span-F1
1	tulbure	0.6129	0.6159	0.6129	0.6108
2	Zleon	0.5875	0.5439	0.6474	0.5939
3	AI_Fusion	0.5777	0.5437	0.6189	0.5843
4	CHEEXIST	0.5621	0.5379	0.5995	0.5456
5	virmel	0.5616	0.4963	0.6584	0.5620
6	miqarn	0.5603	0.5117	0.6273	0.5618
7	DSVS	0.5529	0.5384	0.5785	0.5323
8	TargaMarhuenda	0.5364	0.5128	0.5710	0.5385
9	Elias&Sergio	0.5151	0.4864	0.5533	0.5231
10	hinlole	0.4994	0.4530	0.5740	0.4890
	baseline-BETO	0.4934	0.4533	0.5621	0.4952
11	Dap_upv	0.4914	0.4555	0.5474	0.4917
12	zhengqiaozeng	0.4903	0.4507	0.5494	0.4874
13	ALC-UPV-JD-2	0.4885	0.4509	0.5458	0.4683
14	ezio	0.4869	0.4623	0.5229	0.4947
15	nlpln	0.4672	0.5174	0.4426	0.2961
16	rfenthusiasts	0.4666	0.5104	0.4341	0.4697
17	SINAI	0.4151	0.4630	0.4054	0.4781
18	TheGymNerds	0.3984	0.3621	0.4483	0.5024
19	DiTana	0.3004	0.4490	0.2362	0.3117
20	ROCurve	0.2649	0.2706	0.2627	0.3562
21	TokoAI	0.1878	0.1189	0.5659	0.1739
22	epistemologos	0.1657	0.1906	0.1864	0.1534
23	LaDolceVita	0.1056	0.1158	0.0975	0.1321
24	theateam	0.0994	0.1051	0.0962	0.0358
25	oppositional_opposition	0.0037	0.0349	0.0022	0.0014

Table B.4: Results for task 2 - Spanish [48]

List of Figures

List of Tables

Acknowledgements

A heartfelt thanks goes to my supervisors, Professor Rosso and Professor Carman for their availability and their support.

A special thanks goes to Mariona that guided me through this project and introduced me to the PRHLT research center.

The biggest thanks goes to my mother and father for how they educated me and supported me at all times. Thank you for not allowing me to give up, even when I wanted to because I saw the exams as too difficult to pass.

Like my parents, I must thank Bianca, my sister, who saw more potential in me than anyone else I know and encouraged me to start this university journey. Without you three, all of this would not have been possible. I would also like to dedicate this space to the people who have been with me over the years, even if only for a short period.

A heartfelt thanks also goes to Paula, whose unwavering support and belief in me never faltered. Your encouragement, patience, and understanding throughout this journey have meant more to me than words can express, and I am forever grateful for your presence and care.

Finally, a sincere thank you to all the incredible people i met these years. Our shared experiences, and mutual support have made this journey truly unforgettable. I am grateful for every moment we faced together, and I couldn't have asked for better people along the way.

This work was one of the research activities of the XAI-DisInfodemics project on eXplainable AI for disinformation and conspiracy detection during infodemics, funded by MCIN/AEI/ 10.13039/501100011033 and by European Union Next Generation.