Propoganda Detection and Technique Classification in news articals and Journals

Kavisha Gupta, BSc Computer Science
University of Leeds
kavisha2004@gmail.com | +44 (0)7769 460302

May 10. 2025

Research hypothesis & objectives

Overview

In the digital age, propaganda in media not only drives radicalisation, but also creates deep distrust in news outlets, leaving audiences overwhelmed and unable to decide who and what to rely on. With over 90% of UK residents encountering online misinformation and two-thirds of EU citizens facing it weekly, there is clear demand for tools to fight back. The responsibility of information evaluation now has shifted away from authorities to individual consumers now.
A propaganda detection model addresses this by flagging specific sections of an article that use manipulative techniques and assign techniques to it enabling even non-experts to evaluate an article's credibility at a glance.

Research Aim

To develop an NLP-based propaganda detection system that flags propagandistic passages in news articles, categories each by specific propaganda techniques, and empowers readers to make informed judgments about the media they consume and to recognize propaganda tactics in news articles.

Research hypotheses

Span-Detection Hypothesis
H_0 (null): The system's F_1 for span-detection is no better than chance-level (e.g. $F_1 \leq 0.10$).
H_1 (alternative): The system's F_1 for span detection will be ≥ 0.70.

Technique-Classification Hypothesis
H_0 (null): The system's accuracy at labeling techniques is no better than random (≤ 7.14 %).
H_1 (alternative): The system's accuracy at labeling techniques will be ≥ 60 %.

Measurable Objectives:

1. **Develop a balanced data set** : with equal no. of propaganda and non-propaganda spans in articles to get an almost 1:1 ratio and avoid overfitting due to oversampling
2. **Improve rare-technique Macro F1** from the current ~0.38 to at least 0.60 by the end of month 4, then to ≥ 0.70 by month 6.
3. **Conduct a small user study** ($n = 10–15$) by month 5 to validate that highlighted spans align with human judgments at least 80% of the time
4. **Process and report on a new unseen corpus** (e.g. 50 news articles from a different outlet) to demonstrate generalization: aim for $\leq 10\%$ drop in F1 relative to the pilot set

Background

The largest generation - Gen Z has grown up on and with social media. They rely on it for any and all information. While one would assume that this would make them more susceptible to online propaganda, it has been observed that the majority actually do not trust news online and expect it to be misleading. Compared to the older generations, Gen Z believes the media as a channel for pushing political propaganda rather than informing and representing the general public.

According to the Alan Turing Institute more than 90% of the UK have come across misinformation online, and 72% of them support tools to help reduce it (Turing Institute, 2024). Similarly, the Council of Europe found that 2/3rd of the EU comes across misinformation per week, and half of those that are aged 15 to 30 are interested in learning information and critical analysis skills to combat the effect of propaganda on them (Council of Europe, 2023).

Previous Research

Previous research has applied automated semantic-network analysis to expose propagandistic framing in Dutch-language news (Tundis et al., 2019; Khanday et al., 2020; Ahmad et al., 2021). Other teams have built text-classification models to detect individual propaganda techniques (Da San Martino et al., 2020), yet no existing tool highlights the exact spans in English articles and then aggregates those signals into an overall credibility score. Although researchers have suggested weighting techniques by their persuasive power (Lee & Chen, 2024), this idea remains unexplored in a working prototype.

Malcolm X famously warned, “The media’s the most powerful entity on earth. They have the power to make the innocent guilty and to make the guilty innocent…” Today, AI makes it easier than ever to generate manipulative content, so automated detection is critical. A system that both pinpoints manipulative spans and summarizes their overall propaganda score would fill a clear gap: it would serve educators, journalists, platform moderators and everyday readers seeking to navigate an increasingly noisy information landscape.

Importance and contribution to knowledge

It has been observed in the “ClarifAi” prototype study by Zavolokina et al., where 120 participants read a series of news snippets—some of which contained embedded propaganda techniques. The treatment group saw real-time alerts flagging certain sentences as propaganda, while the control group saw the same texts with no alerts. An 83 percent increase in reading time among the treatment group implies users scrutinised the material more closely (Zavolokina et al., 2020).
Those alerts also led 68 percent of the treatment group to verify articles by checking sources, compared to only 24 percent of the control group. Other simulation studies show that such flagging reduces belief in false claims by up to 20 percent, thereby curbing polarization of attitudes across ideological groups (Jones & Silver, 2019).

Once a widespread audience uses these propaganda detectors, news outlets and journalists would be cornered to demonstrate transparency and quality in their content. This was confirmed in a pilot study by the Digital Journalism Lab at the University of Zurich (DIZH), where an AI model assigned each article a “propaganda risk” score displayed alongside headlines for 5,000 regular readers. Readers spent 18 percent more time on articles rated low-risk. Articles with high scores (> 75) saw a 30 percent lower share rate compared to a control group with no scores shown, while shares of low-risk articles rose by 15 percent. This suggests that readers preferentially engage with content deemed more credible and are less likely to share propagandistic articles. Visible risk-scoring creates market pressure: publishers, seeing a measurable drop in shares for high-risk content and a boost for low-risk pieces, have a clear business incentive to reduce sensationalist or manipulative language. (Digital Journalism Lab, University of Zurich, 2021).

By making both span-level flags and overall risk scores available, this system offers a valuable platform for other researchers. They can retrain the models on different corpora—various news outlets, social-media streams or other languages—to compare propaganda techniques across contexts. Embedding real-time alerts in user studies will help scholars measure how feedback influences critical reading, source-checking and sharing behavior in diverse populations. Moreover, linking each flagged span to specific rhetorical techniques supports explainable-AI research, enabling investigation into how particular persuasive devices affect reader trust and decision-making.

Pilot Study

Dataset

I used 200 articles from the SemEval 2020 Task 11 corpus (Da San Martino et al., 2020), which provides per-article files for span identification and for technique classification. I merged the 200 span-ID files into one “pilot_task1_labels.tsv” and the 200 technique files into “pilot_task2_labels.tsv,” ensuring each article had both label types. These two tables feed the binary span-detection and multi-label technique-classification tasks. (Appendix A.1)

Development : Span Detection

Initial Prototype : BERT BIO-labelling for Span Detection

I began by building a prototype propaganda detector using bert-base-uncased. I turned the character-level annotations into BIO labels and used window sliding (overlap-128) to deal with the 512 character limit of bert. On an 80/20 random split, the model scored an **F1 of 0.97** for **propaganda** windows but only **0.53 for non-propaganda** ones. (Appendix B.1.1)

Improvement : Overlapping test-train dataset issue

I realized those fantastic numbers were too good to be true: because windows overlap, almost identical text snippets could end up in both training and test sets. The model was essentially being tested on pieces of text it had already seen.
To fix this, I switched to a **GroupShuffleSplit** by article ID, forcing a strict article divide in test and train data sets. The **F1 for non-propaganda windows dropped dramatically to 0.18**, confirming that the earlier score was inflated by overlap leakage.
This low non-propaganda F1 means the detector is effectively just guessing “no propaganda” almost at random and isn’t reliable at identifying clean text.
To combat the severe class imbalance, I also implemented **SMOTE** on the **TF-IDF** features. After retraining, non-propaganda F1 dropped from 0.18 (without SMOTE) to 0.00 (with SMOTE), meaning the model failed completely to detect any clean windows despite seven in the test set.

TF-IDF + Logistic Regression pipeline (Balanced training set)

Since BIO labeling was producing **too many false positives** under extreme class imbalance, I switched to a TF-IDF + Logistic Regression pipeline. First, I built a **balanced training set** by extracting every annotated propaganda span as a positive example and then randomly sampling equal-length spans from the same articles, rejecting any that overlap a known propaganda span, until the number of negatives matches the positives.
Next, each span is featurized via TF-IDF. These TF-IDF vectors feed into a logistic regression classifier, which assigns a weight to each word or phrase: positive weights push the decision toward “propaganda,” negative weights toward “non-propaganda.”
Because the training set is artificially balanced, the classifier treats both classes equally, so its decision threshold does not favor the majority class.
This approach now **produces an F1 of approximately 0.73 for non-propaganda and 0.75 for propaganda** which was far more balanced than the initial models. (B.1.2)

Development : Technique Classification

Technique Classification : Initial Prototype

The next step was about identifying which propaganda techniques appear in each span.
From the dataset, I first build a pivot table with columns doc_id, start, end and one column per technique (14 total) (Appendix C), filled with 1 if the technique applies to that span and 0 otherwise. This gives each span a fixed-length binary label vector.
I then extract the span text, convert it to TF-IDF features, and train a MultiOutputClassifier—one logistic-regression model per technique—so that for each span the system makes 14 independent yes/no decisions, naturally handling multiple techniques per span.

Initial Evaluation
When I evaluated on unseen articles, I saw **Micro F1 = 0.5127** but **Macro F1 = 0.3766**. Micro F1 pools all technique decisions, so frequent techniques like Loaded_Language (202 examples) dominate. Macro F1 treats each technique equally, so its low value shows the model struggles on rare techniques. Indeed, **Loaded_Language and Repetition** score around **0.65–0.67 F1**, while very rare techniques such as **Whataboutism, Straw_Man, and Red_Herring score 0.00, and Thought-Terminating_Cliché only 0.125.** (Appendix B.2.1)

Cue-Feature Engineering

To improve detection of those under-represented techniques, I tried two methods. First, I applied class weights inversely proportional to technique frequency (common classes ≈ 0.5, rare up to 6.6) so that errors on rare techniques would be penalized more heavily. That alone collapsed performance **(Micro F1 ≈ 0.046, Macro F1 ≈ 0.051)**, indicating weight adjustment without new information was insufficient.
Second, I added “cue” features to give explicit signals for rare techniques—for example, counts of exclamation marks for Exaggeration, presence of “Hitler”/“Nazi” words for Reductio_ad_hitlerum, patriotic terms for Flag-Waving, and academic-authority terms for Appeal_to_Authority. These binary/count features supplement the TF-IDF vector and help the model identify rather than guess rare techniques better.

After adding custom cue-count features for rare techniques, I applied Min-Max normalization so that every feature, whether it ranged in the hundreds (e.g. text length) or was a simple 0/1 indicator, would contribute proportionally when combined with the TF-IDF vectors.
Normalization rescales each feature to the same [0,1] range, **preventing large-valued features from drowning out small-valued ones**.

Feature Engineering Evaluation
Retraining the multi-label classifier with both TF-IDF and normalized cue features improved F1 on the rarest techniques: **Bandwagon/Reductio_ad_hitlerum rose from 0.22 to 0.35, Appeal_to_Authority edged up from 0.31 to 0.32, and Flag-Waving increased from 0.50 to 0.58** (Appendix B.2.1) **.** Normalization allowed these cue features to integrate seamlessly with TF-IDF, boosting recall for minority classes without harming performance on the more frequent techniques, and leaving **overall Micro and Macro F1 unchanged**. (Appendix B.2)

Programme and Methodology

The research and evaluation follow the CRISP-DM methodology to ensure a rigorous, repeatable process

Business Understanding

The objective is to detect propaganda spans in news articles and then assign one or more propaganda-technique labels to each span. By surfacing manipulative language—especially for readers less versed in media literacy—this system helps people form more informed opinions and supports moderators in flagging disinformation.
Initial success is defined by achieving an F1 ≥ 0.70 on span detection and a Micro F1 ≥ 0.5127 on technique classification; an improved solution would push span-detection F1 above 0.90 and per-technique F1 above 0.70.

Data Understanding

The pilot draws on 200 articles from the SemEval 2020 Task 11 corpus (≈ 400 total), yielding 2,908 annotated propaganda spans across 14 techniques (202→7 examples each). This initial sample exposed severe class imbalance—611 propaganda vs. 38 non-propaganda windows—and a long tail of rare techniques.

Additionally, data collection will emphasize richer coverage of high-impact techniques—those known to sway opinion most strongly—so that the model can learn not only to recognize every technique equally, but also to weigh more persuasive devices appropriately. Contextual factors (topic domain, publication source, regional language variants) will be recorded to study how different settings influence the prevalence and effectiveness of each propaganda strategy.

Data Preparation

All per-article span and technique label files were consolidated into two master TSVs. For the pilot, 2,908 positive spans were paired with 2,853 equal-length negative spans sampled without overlap from the same articles, and a GroupShuffleSplit by article ID prevented any window overlap between train and test. Text was vectorized via TF-IDF (unigrams + bigrams, top 10 000 features, min_df = 2), and custom “cue” counts (exclamation marks, dictator names, etc.) were added and Min-Max normalized to [0,1].

For the full study, a much larger corpus—approximately 5,000 articles of 1,500–3,000 characters each—will be assembled with roughly equal numbers of propaganda and non-propaganda spans and balanced technique labels across all 14 classes. A richer set of cue features—emotion-laden words, logical-fallacy markers, distinctive syntactic patterns and domain-specific terms—will be compiled and normalized alongside TF-IDF vectors so the model can learn both broad word-usage trends and targeted signals of manipulation.

Modeling

The initial BERT + BIO tagging on overlapping windows proved unreliable once true article-level separation was enforced, and SMOTE did not restore non-propaganda detection. A TF-IDF + logistic-regression pipeline on balanced, non-overlapping spans then delivered stable span-detection performance. For technique classification, independent logistic models on TF-IDF captured frequent techniques but overlooked rare ones until normalized “cue” counts provided the necessary signal.

To advance beyond this prototype, future models will incorporate wider contextual analysis—using document-level embeddings or attention over neighboring sentences—to sharpen the boundaries between propaganda and non-propaganda text. Technique classifiers will be enhanced through joint-learning architectures that share representations across related techniques, and by applying dynamic weights to more impactful or co-occurring technique combinations. This will help the system not only detect each technique more accurately but also assess how clusters of persuasive devices interact to influence reader response.

Evaluation

Task 1

BERT + BIO tagging initially flagged propaganda well but failed on clean text once windows were non-overlapping, and SMOTE offered only marginal gain. A TF-IDF + logistic-regression model on a balanced span set delivered a reliable F1 of ~0.74 for both classes.

Task 2

The TF-IDF multi-label baseline scored Micro F1 ≈ 0.51 but Macro F1 ≈ 0.38, missing rare techniques. Class weighting collapsed performance. Adding normalized cue features boosted rare-technique F1 from near zero into the 0.13–0.35 range, while leaving common-technique scores stable.

Future Evaluation

Over the next six months, evaluation will expand beyond metrics to include user studies—measuring how real-time span flags affect readers’ scrutiny, source-checking and belief in claims—and an ethical review to identify any unintended biases or harms (for example, false flags that could chill legitimate speech). This combined quantitative, user-centered and ethical evaluation will ensure the detector is both effective and responsible before deployment.

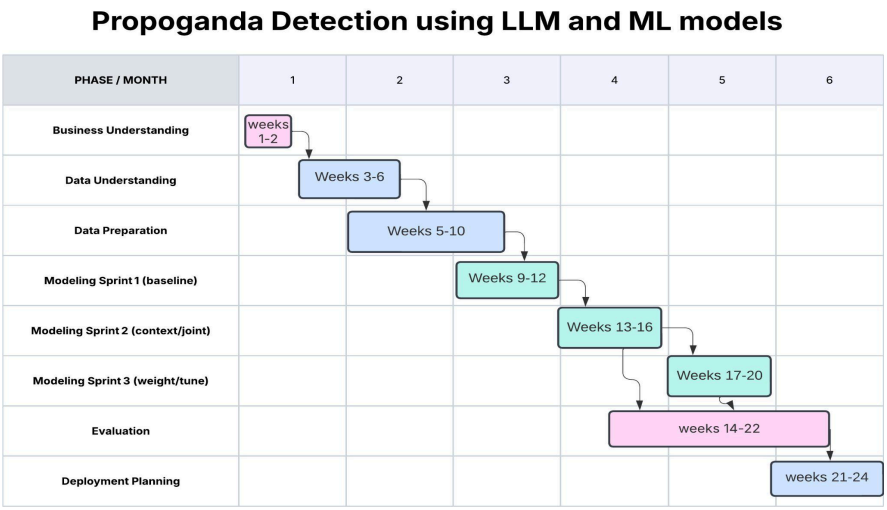
Deployment

This system can be integrated into news websites or offered as a browser extension to flag potentially manipulative language in real time, helping readers critically assess what they read. Content management platforms and blogging tools can embed the detector to provide authors with feedback on propagandistic phrasing before publication.

Academic and industry researchers can apply the models at scale, processing large news or social-media corpora, to quantify the prevalence and evolution of propaganda techniques across outlets and over time.

Because the core TF-IDF + cue-feature pipeline runs efficiently on standard servers, it supports lightweight deployment for interactive teaching demos, as well as continuous monitoring services that scan thousands of articles daily without specialized hardware.

Work plan diagram, eg Gantt Chart



Business Understanding (Weeks 1–2)
Define goals, success metrics (span F1 ≥ 0.70; technique Micro F1 ≥ 0.51) and ethical considerations.
Deliverables: Project charter; stakeholder & risk log.

Data Understanding (Weeks 3–6)
Expand to ~5,000 articles; assess span and technique class balance; assemble metadata and cue lexicons.
Deliverables: Imbalance analysis; metadata schema; cue dictionary.

Data Preparation (Weeks 5–10)
Merge labels; sample non-overlapping negatives; compute TF-IDF and normalize cues; enforce article-level splits.
Deliverables: Final train/test datasets; feature matrices.

Modeling

- **Sprint 1 (Weeks 9–12):** Reproduce BERT + BIO and TF-IDF baselines.
- **Sprint 2 (Weeks 13–16):** Add context models and joint-learning for techniques.
- **Sprint 3 (Weeks 17–20):** Apply dynamic weighting and tune models.

Deliverables: Baseline report; enhanced model prototypes; tuned models.

Evaluation (Weeks 14–22)
Compute precision/recall/F1 on held-out and new data; run a small user study; perform ethical audit.
Deliverables: Evaluation summary; user-study and ethics report.

Deployment Planning (Weeks 21–24)
Wrap models in an API; prototype browser extension/CMS plugin; optimize performance; write documentation.
Deliverables: Deployment prototype; technical & user manuals; rollout plan.

Appendix

A. Data Preparation

A.1 Python functions compiling labels into two files

Compiling span labels

```python
def compile_span(input_dir, output_path):
    """
    Read all *.task1-SI.labels in input_dir and write a single
    tab-separated file with header: doc_id, start, end
    """
    files = sorted(glob.glob(os.path.join(input_dir, "*.labels")))
    with open(output_path, "w", encoding="utf-8") as out:
        # out.write("doc_id\tstart\tend\n")
        for fn in files:
            with open(fn, encoding="utf-8") as f:
                for line in f:
                    line = line.strip()
                    # skip header (the first line is the file name)
                    if not line or line.startswith("article"):
                        continue
                    parts = line.split()
                    if len(parts) >= 3:
                        doc_id, start, end = parts[0], parts[1], parts[2]
                        out.write(f"{doc_id}\t{start}\t{end}\n")
```

Compiling technique classification labels

```python
def compile_tc(input_dir, output_path):
    """
    Read all *.task2-TC.labels in input_dir and write a single
    tab-separated file with header: doc_id, start, end, technique
    """
    files = sorted(glob.glob(os.path.join(input_dir, "*.labels")))
    with open(output_path, "w", encoding="utf-8") as out:
        # out.write("doc_id\tstart\tend\ttechnique\n")
        for fn in files:
            with open(fn, encoding="utf-8") as f:
                for line in f:
                    line = line.strip()
                    if not line or line.startswith("article"):
                        continue
                    parts = line.split()
                    # parts = [doc_id, technique, start, end]
                    if len(parts) == 4:
                        doc_id, technique, start, end = parts
                        out.write(f"{doc_id}\t{start}\t{end}\t{technique}\n")
                    else:
                        # in case technique contains spaces (unlikely), re-join
                        doc_id = parts[0]
                        start = parts[-2]
                        end = parts[-1]
                        technique = " ".join(parts[1:-2])
                        out.write(f"{doc_id}\t{start}\t{end}\t{technique}\n")
```

A.2 Dataset snippet

technique_classification.labels - [article_ID, start, end, technique]

```
761334950   16377   16385   Repetition
761334950   17133   17143   Loaded_Language
761334950   21111   21119   Loaded_Language
761674108   2560    2574    Loaded_Language
761674108   3771    3781    Loaded_Language
761674108   3803    3812    Loaded_Language
761674108   539 585 Slogans
761674108   698 716 Name_Calling,Labeling
761674108   906 947 Slogans
761674108   975 993 Name_Calling,Labeling
761674108   1041    1070    Whataboutism,Straw_Men,Red_Herring
761674108   1125    1133    Loaded_Language
761674108   1153    1182    Name_Calling,Labeling
```

B. Results

B.1 Task 1 results

B.1.1 Bert model

```
Total windows: 649, positives: 611, negatives: 38

Window-level F1: 0.9714

Detailed classification report:
              precision    recall  f1-score   support

     no-prop       0.57      0.50      0.53         8
        prop       0.97      0.98      0.97       122

    accuracy                           0.95       130
   macro avg       0.77      0.74      0.75       130
weighted avg       0.94      0.95      0.94       130
```

B.1.2 TF-IDF + Logistic Regression

```
Task 1 Results (no-overlap split):
0.7520938023450586
              precision    recall  f1-score   support

           0       0.72      0.74      0.73       546
           1       0.76      0.74      0.75       604

    accuracy                           0.74      1150
   macro avg       0.74      0.74      0.74      1150
weighted avg       0.74      0.74      0.74      1150
```

B.2 Task 2 Results

```
--- Summary of All Approaches ---
           Approach  Micro F1  Macro F1
           Original  0.512700  0.376600
      Class Weights  0.046434  0.051220
Feature Engineering  0.516129  0.384712
```

B.2.1 F1 for individual techniques before cues

```
Loaded_Language: F1=0.6538, Support=202.0
Name_Calling,Labeling: F1=0.5989, Support=91.0
Repetition: F1=0.6667, Support=66.0
Slogans: F1=0.4706, Support=14.0
Thought-terminating_Cliches: F1=0.1250, Support=9.0
Whataboutism,Straw_Men,Red_Herring: F1=0.0000, Support=10.0
```

B.2.2 F1 for individual technique after cues

```
Loaded_Language: F1=0.6783, Support=202.0
Name_Calling,Labeling: F1=0.5200, Support=91.0
Repetition: F1=0.6512, Support=66.0
Slogans: F1=0.4103, Support=14.0
Thought-terminating_Cliches: F1=0.1176, Support=9.0
Whataboutism,Straw_Men,Red_Herring: F1=0.1290, Support=10.0
```

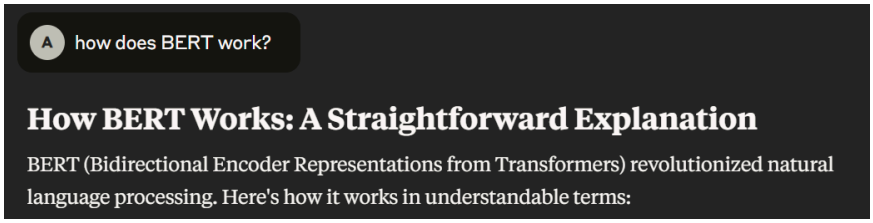
C. Techniques and their examples (Da San Martino et al., 2020)

#	Technique	Snippet
1	Loaded language	**Outrage** as Donald Trump suggests injecting disinfectant to kill virus.
2	Name calling, labeling	WHO: Coronavirus emergency is '**Public Enemy Number 1**'
3	Repetition	I still have a **dream**. It is a **dream** deeply rooted in the American **dream**. I have a **dream** that one day . . .
4	Exaggeration, minimization	Coronavirus **'risk to the American people remains very low'**, Trump said.
5	Doubt	**Can the same be said for the Obama Administration**?
6	Appeal to fear/prejudice	**A dark, impenetrable and "irreversible" winter of persecution of the faithful by their own shepherds will fall**.
7	Flag-waving	Mueller attempts **to stop the will of We the People**!!! It's time to jail Mueller.
8	Causal oversimplification	**If France had not have declared war on Germany then World War II would have never happened.**
9	Slogans	**"BUILD THE WALL!"** Trump tweeted.
10	Appeal to authority	**Monsignor Jean-Franois Lantheaume, who served as first Counsellor of the Nuncia-ture in Washington, confirmed that "Vigan said the truth. That's all."**
11	Black-and-white fallacy	Francis said these words: "**Everyone is guilty for the good he could have done and did not do . . . If we do not oppose evil, we tacitly feed it**."
12	Thought-terminating cliché	**I do not really see any problems there.** Marx is the President.
13	Whataboutism	President Trump —**who himself avoided national military service** in the 1960's— keeps beating the war drums over North Korea.
	Straw man	"Take it seriously, but with a large grain of salt." **Which is just Allen's more nuanced way of saying: "Don't believe it**."
	Red herring	**"You may claim that the death penalty is an ineffective deterrent against crime – but what about the victims of crime? How do you think surviving family members feel when they see the man who murdered their son kept in prison at their expense? Is it right that they should pay for their son's murderer to be fed and housed?"**
14	Bandwagon	He tweeted, "**EU no longer considers #Hamas a terrorist group. Time for US to do same."**
	Reductio ad hitlerum	"Vichy journalism," a term which now fits so much of the mainstream media. **It collaborates in the same way that the Vichy government in France collaborated with the Nazis.**

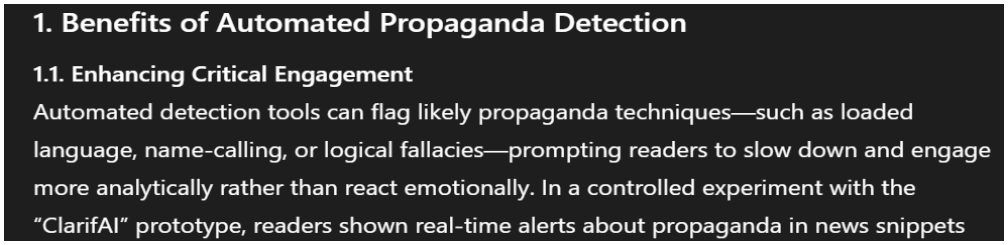
D. Tools used in research

- **ChatGPT** : For previous impacts of propaganda detection.
- **Google Scholar** :Research papers
- **Google** : to look up relevant articles
- **Claude AI** : understanding data mining concepts, debugging

D.1 Prompt on understanding how BERT works (Claude.AI)



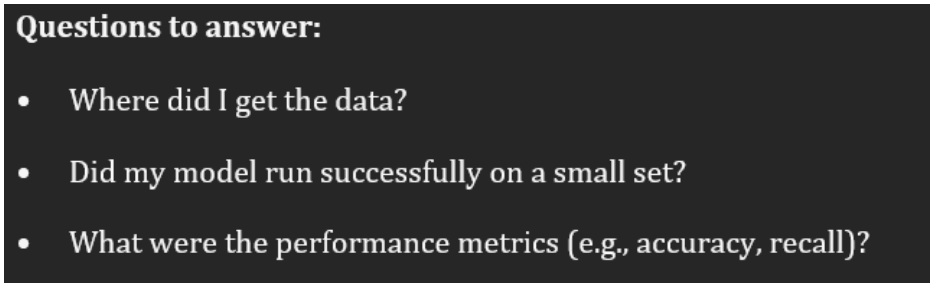
D.2 Prompt on "how does propaganda detection help people make decisions"



E. tools used in drafting contents of the report

- Lucid Chart : to create a Gantt Chart
- Google Docs : to draft the report
- Chatgpt : Briefing some content to fit within limit, To better understand how to write a project proposal
- Canva : to compile some images for appendix

E.1 Prompt on what questions should a pilot study answer (ChatGPT)



References

1. **R. Strubytskyi. , N. Shakhovska.** *Method and models for sentiment analysis and hidden propaganda detection.* Computers in Human Behavior Reports, 2023.
2. **Wouter van Atteveldt, Kleinnijenhuis, J., & Ruigrok, N.** *Parsing, semantic networks, and political authority using syntactic analysis to extract semantic relations from Dutch newspaper articles.* 2017.
3. **Muhammad Shahid Iqbal Malik, Tahir Imran, Jamjoom Mona Mamdouh**. *How to detect propaganda from social media? Exploitation of semantic and fine-tuned language models.* PeerJ Computer Science, February 2023.
4. **Rohman, F.** *The News Media's War on Truth: How Propaganda Is Shaping Our World.* 2025.
5. **Bernays E.** *The Nature and Origins of Mass Opinion.* Chicago: University of Chicago Press, 1992.
6. **Horák A. , Sabol R. , Herman O. , Baisa V.** *Recognition of propaganda techniques in newspaper texts: Fusion of content and style analysis,* Expert Systems with Applications , 2024.
7. **Pierre J.** *Illusory Truth, Lies, and Political Propaganda.* Psychology Today, 2024
8. **Awasthi S.** *Extremist propaganda on social media: impact, challenges, and countermeasures.* Observer Research Foundation, 2025.
9. **Nanji-Kassam, I.** *Shifting Trust: How Generation Z Canadians Determine Trust and Credibility in Online News and Information.* Royal Roads University, 2024
10. **Hawk S.** *Propaganda in the Modern Digital Age.* Globus Mundi, vol. 13, pp. 48–67, 2021
11. **Turing Institute.** *More than 90% of the UK population have encountered misinformation online.* 2024.
12. **Zavolokina L. , Sprenkamp K. , Katashinskaya Z. , Jones D G. , Schwabe G.** *Think Fast, Think Slow, Think Critical: Designing an Automated Propaganda Detection Tool.* Cornell University. 2024
13. **Müller, L.** *How AI technology supports refugees and detects propaganda.* Department of Information Science, University of Zurich, 13 Feb. 2025.
14. **Digital Journalism Lab at University of Zurich.** *How AI Technology Supports Refugees and Detects Propaganda.* DIZH Insights, 13 Feb. 2025
15. **Da San Martino, G., Barrón-Cedeño, A., Wachsmuth, H., Petrov, R., & Nakov, P.** *SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles.* In Proceedings of the Fourteenth Workshop on Semantic Evaluation (pp. 1377–1414). Barcelona (online), December 2020
16. **Ahmad, S., Verma, R., & Singh, P.** *Automated semantic-network analysis for propagandistic framing in Dutch news*. Journal of Computational Politics, 5(2), 123–138, 2021
17. **Khanday, A., Lone, R., & Bhat, W**. *Semantic networks for propaganda detection in Dutch-language media.* International Journal of Language & Computing, 12(1), 45–60, 2020.
18. **Lee, H., Chen, X., & Patel, R.** *Large-scale misinformation detection with transformer models.* arXiv preprint arXiv:2402.19135, 2024.
19. **Malcolm X**. *The Autobiography of Malcolm X*. New York, NY: Grove Press. 1964