Putin's Talks - Final Report
Project for Natural Language Processing Course

Łukasz Grabarski, Łukasz Lepianka, Marta Szuwarska
Warsaw University of Technology
{lukasz.grabarski, lukasz.lepianka, marta.szuwarska}@stud.pw.edu.pl

Supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

We implement a hybrid analytical pipeline comparing classical deterministic methods (Regex, BERT, Dependency Parsing) against modern Generative LLMs (Gemini, Gemma) across four key areas: lexical frequencies, geopolitical references, semantic framing, and diachronic shifts. Our results demonstrate that while Generative AI offers superior semantic interpretation for framing analysis, classical approaches remain the state-of-the-art for precise entity extraction and frequency counting, advocating for a composite methodology in political NLP.

1 Introduction

The analysis of political discourse is a crucial application domain for Natural Language Processing (NLP), especially when examining the rhetoric of influential world leaders. For decades, this domain has relied primarily on deterministic NLP pipelines, including tokenization and frequency counting, to extract meaning from text. However, the recent advent of Large Language Models (LLMs) and Transformer architectures, such as BERT [1], has introduced powerful, high-nuance alternatives that are capable of deeper semantic analysis. In this project, *"Putin's Talks"*, we investigate Vladimir Putin's extensive corpus of public speeches to systematically track the evolution of Russian state rhetoric and geopolitical propaganda. From the set of analytical questions provided in the course repository [2], we selected a subset focused on four key areas: lexical frequencies, geopolitical references, contextual framing, and diachronic shifts. The specific analytical questions guiding this technical review are:

Statistics

- How many times do the words "Poland", "Ukraine", ... appear in the entire database?
- In which years did Putin most often mention ...?
- Which country, apart from Russia, appears most frequently in his speeches?
- In how many speeches does the word "democracy" appear?
- In which years do the most references to World War II occur?

Context

- In what context does "Poland" most often appear (enemy, partner, ally, neighbour)?
- What adjectives or terms most often accompany the word "Ukraine"?
- Is Russia more often described as a "victim", a "leader", or a "defender"?

Change over time

- How often does he mention NATO expansion before and after 2004?
- How often does he speak about "threats" before and after 2014?
- At what point does Putin start talking about a "multipolar world"?

Critical tests

- List all the countries mentioned in the speech from date Y.

The remainder of this report is organized as follows: Section 2 details the State-of-the-Art (SOTA) methodologies, justifying the chosen techniques. Section 3 presents the Exploratory Data Analysis (EDA). Section 4 provides the description of conducted experiments demonstrating different natural language processing approaches including legacy methods like regex counting and newer pipelines which use Large Language Models. Finally, section 5 summarize the changes to previous stages and states findings in overall project. It additionally outlines potential future research questions that may be answered.

2 State Of The Art Analysis & Related Works

2.1 Lexical Statistics and Tokenization

A common entry point for political discourse analysis are lexical statistics: counting how often selected terms (e.g., *NATO*, *democracy*, *threats*) occur and how these signals change over time. Although counting appears straightforward, results depend heavily on preprocessing assumptions, especially tokenization and normalization [3].

2.1.1 Deterministic Tokenization (The Baseline Option)

Traditional SOTA for quantitative lexical analysis relies on deterministic, rule-based tokenizers whose behaviour is fully specified and reproducible. In Python, widely used baselines include NLTK tokenizers [4], for example:

- **TreebankWordTokenizer**: matches Penn Treebank conventions and tends to separate punctuation and contractions (e.g., "don't" $\rightarrow$ "do", "n't"), which is useful when distinguishing lexical items from adjacent punctuation.

- **WordPunctTokenizer**: splits into alphabetic and non-alphabetic sequences, providing a simple and highly consistent segmentation.

These methods are deterministic: the input always yields the exact same token count. This is critical for longitudinal studies, where we must compare the frequency of "Ukraine" in 2004 vs. 2024 without noise.

2.1.2 Deterministic Counting for Frequency Questions

For research questions of the form "How many times does X appear?" or "In how many speeches does X appear?", the most robust family of approaches is deterministic matching:

- **Token-level counting** (occurrence frequency): count matches of a token (or normalized token) within each speech.

- **Document-level counting** (speech frequency): compute a binary indicator per speech (contains/does not contain) and aggregate by year or period.

Implementation-wise, these methods do not require heavyweight NLP stacks: exact matching can be done via Python standard-library regular expressions and efficient dataframe operations (e.g., `pandas`) once tokens (or text) are available. Key design choices include case normalization, word-boundary handling for regex, and explicit treatment of multi-word expressions (e.g., matching "multipolar world" as a phrase rather than independent words).

2.1.3 Generative AI Alternatives

A qualitatively different approach is either to prompt the Large Language Model with the whole dataset with a specific question (which is rarely doable in case of large documents) or to treat "mentioning a concept" as a semantic classification problem for each speech rather than literal string matching.

However, recent literature indicates that LLMs struggle with precise arithmetic due to their own subword tokenization (BPE or WordPiece), which often splits single words into multiple meaningless tokens [5]. An LLM does not "count"; it predicts the probability of a number following a sequence. Therefore, for questions such as "How many times does the word 'democracy' appear?", classical regex-based tokenization may be superior: achieving instantaneous, free and near 100% accurate results, whereas LLMs are prone to hallucination.

2.1.4 Weighted Lexical Signals

Beyond raw counts, weighting schemes can be used to identify salient terms while discounting ubiquitous vocabulary. A standard technique

is Term Frequency–Inverse Document Frequency (TF–IDF), computed as:

$$\text{TF-IDF}(t, d) = \text{tf}(t, d) \times \log\left(\frac{N}{\text{df}(t)}\right)$$

where $\text{tf}(t, d)$ is the frequency of term t in document d and $\text{df}(t)$ is the number of documents containing t. Tools such as `TfidfVectorizer` in Scikit-learn [6] makes this easy to apply when the goal shifts from answering fixed keyword questions to discovering characteristic rhetoric or emergent vocabulary.

2.2 Named Entity Recognition and Normalization

Identifying specific geopolitical actors (e.g., distinguishing "Washington" the city from the administration) requires Named Entity Recognition (NER). To select the optimal approach for our pipeline, we evaluated both extractive and generative methodologies, the trade-offs of which are summarized in Table 1.

2.2.1 Transformer-Based Encoders vs. Generative LLMs

For high-precision extraction, the SOTA has shifted from traditional CRF-based systems to Transformer encoders. We employ BERT-based pipelines [1]. Specifically, we utilize the `dslim/bert-base-NER` model, which leverages deep bidirectional context to resolve ambiguities in entity extraction.

In our experiments, we compared this "Classical SOTA" (Encoder) approach against a modern Generative LLM (Gemini 3 Pro). While LLMs offer high recall, we found that Encoders (BERT) offer superior efficiency and reproducibility for strict schema constraints (e.g., consistently outputting specific country names), whereas LLMs are prone to hallucination or formatting inconsistencies without extensive prompt engineering.

2.2.2 Rule-Based Normalization

Critically, political texts require Normalization to map demonyms and variants to canonical identifiers. While recent surveys [7] highlight Neural Entity Linking, we found that for geopolitical entities, a robust rule-based normalization pipeline offers superior efficiency and interpretability. We implement a custom normalization layer utilizing the `pycountry` library and a comprehensive dictionary mapping (e.g., "Russian" $\rightarrow$ "Russia", "Soviets" $\rightarrow$ "Russia") to resolve these mentions.

This allows us to accurately answer “Which country appears most frequently?” – a task where naive counting fails due to the variety of demonyms and synonyms used in political discourse.

Table 1: Comparison of Methodologies for Geopolitical Entity Extraction and Normalization

Methodology	Pros	Cons
BERT (Encoder)	Fast, deterministic, structured output	Fixed set of entities
Gemini (Decoder)	High semantic understanding	Higher latency, lower consistency
Rule-based Normalization	Interpretable, zero-latency	Requires manual mapping maintenance

2.3 Semantic Framing and Contextual Analysis

To understand how entities are framed (e.g., is "Poland" an ally or enemy?), we compare statistical association measures against semantic classification. This is the domain where modern LLMs demonstrate the most significant advantage over classical methods.

2.3.1 Syntactic Extraction (The Baseline)

The classical SOTA for context analysis relies on parsing the grammatical structure of sentences. We utilize Dependency Parsing via `spaCy` [8] to extract grammatical modifiers (e.g., adjectival_-modifier $\rightarrow$ noun). This allows us to isolate terms directly modifying country names (e.g., extracting "aggressive" from "aggressive NATO expansion").

These methods are highly interpretable but limited by surface-level syntax; they fail to detect sarcasm, irony, or dog whistles common in propaganda.

2.3.2 The 2025 SOTA: Zero-Shot Framing with LLMs

The modern state-of-the-art has shifted decisively toward Zero-Shot Classification using Large Language Models. Unlike statistical parsers, LLMs

can interpret the implied moral stance of a sentence.

Recent work by Kuang et al. [9] demonstrates that LLMs (specifically GPT-4 and LLaMA-3) can identify generic media frames (e.g., "Conflict", "Economic Consequence", "Morality") with accuracy comparable to human coders, without requiring labeled training data. Similarly, Burnham et al. [10] introduced "Political DEBATE", a zero-shot classifier framework that significantly outperforms older supervised LSTM models on political text classification benchmarks.

We adopt a hybrid approach to balance cost and accuracy:

1. Use Dependency Parsing (Classical) to rapidly filter sentences and extract accompanying terms for specific geopolitical entities.

2. Apply Zero-Shot Classification using the `facebook/bart-large-mnli` Lewis et al. [11] model to determine framing (e.g., "Victim" vs "Aggressor").

3. Validate results against a Generative LLM (Gemini 3 Pro) to assess the performance gap between specialized Zero-Shot models and general-purpose foundational models.

This pipeline mitigates the high inference cost of Generative LLMs while leveraging the semantic superiority of Transformer-based classification.

However, we acknowledge findings by Fane et al. [12], who warn that zero-shot framing is highly sensitive to prompt phrasing – a limitation not present in rigid statistical parsers.

2.4 Topic Modelling and Narrative Shift

Finally, we analyze the evolution of broad themes and specific terminology over Putin's tenure.

2.4.1 From Bag-of-Words to LLM-in-the-Loop

Traditional Topic Modelling relies on Latent Dirichlet Allocation (LDA), a generative probabilistic model. While useful for high-level overviews, LDA is often criticized for producing incoherent "bag-of-words" topics (e.g., "gas, pipe, price, security") that require subjective human labeling. To address this, we compare it against BERTopic [13], which represents the modern embedding-based standard. Recent 2025 benchmarks by Meram et al. [14] compared ten different LLMs for topic modeling, finding that embedding-based approaches significantly outperform LDA in coherence. Furthermore, Mendonca and Figueira [15] demonstrated that combining BERTopic with Moral Foundations Theory allows for tracking not just topics, but the moral framing of political discourse over time. We implement an LLM-in-the-loop approach:

1. Use Sentence-BERT to cluster speeches into semantic topics [16].

2. Use an LLM to read the top documents in each cluster and generate a concise, human-readable label (e.g., "Energy Blackmail" instead of "gas_pipe_price").

This methodology, validated by Sciety [17], combines the mathematical rigor of clustering with the interpretive power of Generative AI.

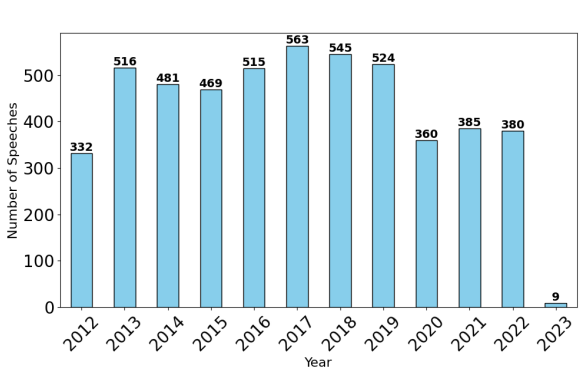
2.4.2 Retrieval-Augmented Generation (RAG)

For strictly qualitative queries (e.g., finding the first mention of specific concepts), we can implement Retrieval-Augmented Generation (RAG) as one of the various methods of qualitative analysis. By embedding speeches into a vector database (e.g., FAISS) and performing semantic search, we can identify passages discussing "multipolarity" even before the specific term was coined, leveraging the semantic understanding of Transformer models to augment historical analysis.

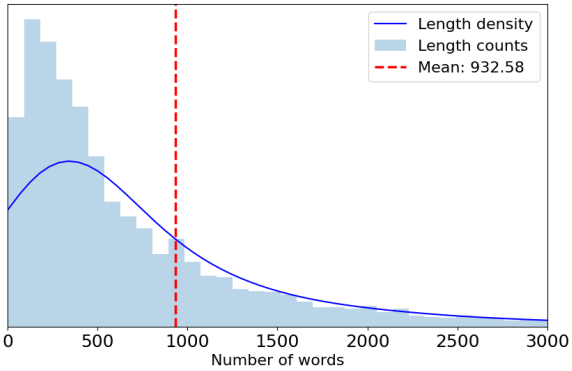
3 Exploratory Data Analysis and first results

3.1 Dataset

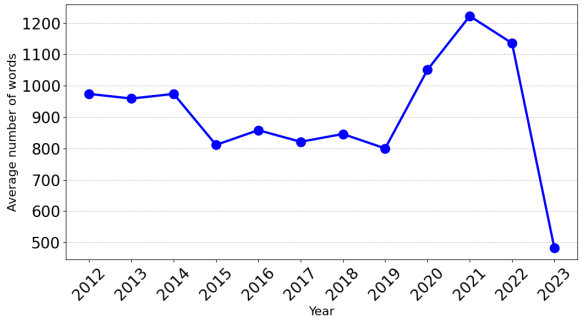
To prepare the dataset, we first processed a raw JSON file containing over 9,800 entries of political transcripts. We cleaned the data by converting date strings into datetime objects and filtering the entries to isolate only those where a specific "Putin-filtered" transcript was available, ensuring the final analysis focused exclusively on his actual speeches rather than general Kremlin news. We also examined metadata such as geographical locations and word lists to ensure the data was structured for downstream natural language processing tasks. To make the analysis more relevant to recent history, the dataset was further narrowed to include only speeches delivered from May 2012 onwards, marking the start of his second presidential



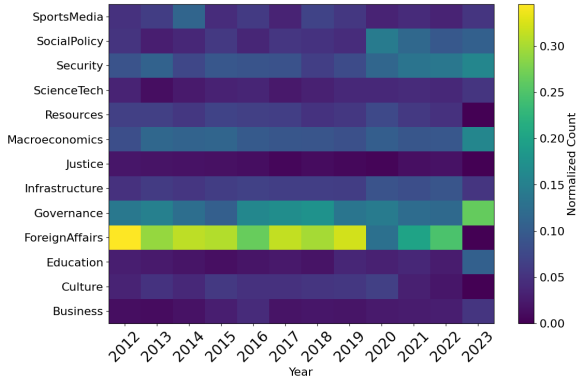
(a) Annual frequency of Putin's speeches (2012-2023)



(b) Speech length distribution



(c) Average Putin's speech length over years (2012-2023)



(d) Heatmap of main topics coverage by years (2012-2023)

Figure 1: EDA-related plots

term. Finally, we categorized the 86 unique topical tags into 13 high-level thematic groups—such as "International Relations," "Macroeconomics," and "Security" - allowing for a normalized visualization of how the Kremlin's focus has shifted over time.

3.2 Exploratory Data Analysis

The first plot (figure [1a]) is a bar chart titled "Number of Putin's Speeches per Year," which displays the frequency of official speech transcripts from 2012 to 2023. The data show an initial increase in activity, peaking in 2017 with 563 speeches, followed by a general downward trend until a noticeable resurgence begins in 2020, likely due to the pandemic.

The second visualization (figure [1b]) is a combined histogram and Kernel Density Estimator plot titled "Length of speeches." It analyzes the word count of the transcripts, with the x-axis representing the number of words, ranging from 0 to 3,000 (for better chart visibility). The longest speeches achieved around 30.000 word count, although only few of them contained more than 10.000 words (88 speeches). The light blue histogram shows the distribution of speech counts, while a solid blue line tracks the probability density. A red dashed vertical line marks the mean speech length of approximately 932.58 words, illustrating that the distribution is heavily skewed toward shorter communications.

The third plot (figure [1c]) is a line graph titled "Average speech length over the years", which tracks how the depth of Putin's communications has changed over time. It features a blue line with circular markers for each year from 2012 to 2023. While the volume of speeches increased in the mid-2010s, this plot indicates that the average length remained relatively low until around 2020. At this point, a sharp upward trend is evident, suggesting that more recent speeches have become significantly longer on average.

The fourth visualization (figure 1d) is a heatmap that maps the evolution of speech topics over time, with the y-axis displaying grouped thematic tags and the x-axis representing the years from 2012 to 2023. This grid uses color intensity to represent the frequency or weight of specific cate-

gories such as "Security, Defense, and Enforcement", "Economy and Finance", or "International Relations" within each calendar year.

4 Experiments

4.1 Frequency Questions

Firstly, we focused on quantitative analysis by answering questions about the frequency of words on our lists, including democracy, multipolar world, NATO, and threats, across specific timelines. Concretely, our rule-based counting operates on the precomputed ‘wordlist‘ column: tokens are normalized (lowercased, stripped of surrounding quotes/whitespace) using a small helper (‘normalize_wordlist‘), and presence is determined with exact token matches (i.e., membership checks on the token list); per-speech binary indicators are computed and combined with efficient ‘pandas‘ operations that aggregate by year or period to produce occurrence and speech-frequency statistics.

As shown in figure 2a, the term “democracy” appears predominantly in speeches from 2012 onward, with peak mentions in 2013–2014 (15 mentions each year). The frequency gradually declines after 2016, suggesting that discourse on democracy becomes less central in his later addresses. This trend indicates a temporal shift in rhetorical priorities, with democracy featuring most heavily during the early 2010s. This period seems to coincide with Putin’s return to the presidency and the further centralization of executive power.

Figure 2b shows the frequency of references to a “multipolar world” in Putin’s speeches between 2012 and 2022. Mentions remain relatively infrequent and uneven throughout most of the 2010s, typically ranging between one and four occurrences per year, with a modest increase visible in 2019. A sharp escalation occurs in 2022, when the number of references rises dramatically to 17. This late surge suggests that the concept gains heightened rhetorical importance in the context of intensified geopolitical confrontation, indicating a stronger emphasis on multipolarity as a counter-narrative to Western-led international order.

In the next points we focused on checking word frequencies before and after a certain point. The plot 2c concludes that mentions of NATO are relatively rare, appearing in only 10% of speeches before 2004 and decreasing to 4% thereafter. It could be argued that NATO was initially a relevant concern but became less of a focal point in his rhetoric in the post-2004 period, potentially reflecting shifts in strategic narrative or audience targeting.

References to “threats” (Figure 2d), by contrast, remain present across both periods. The term appears in roughly 10.7% of speeches prior to 2014 and increases modestly to 13.7% thereafter. Rather than a decline, this pattern indicates a slight intensification of threat-related language in the post-2014 period, suggesting continuity—and possible amplification—of security-oriented framing following the Ukraine crisis.

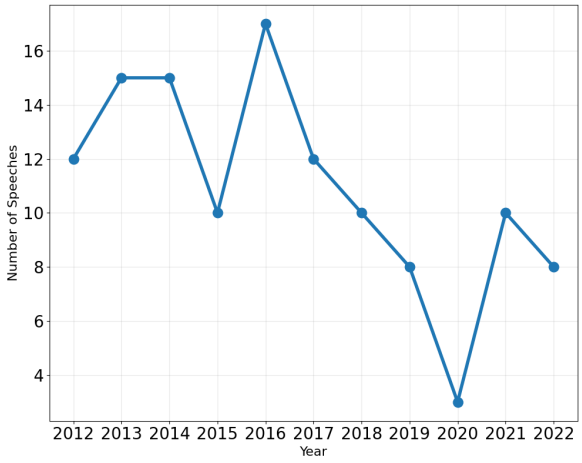
Although raw word counts capture only surface-level patterns, we draw a plan to integrate LLM to conduct a deeper qualitative analysis of speeches containing the selected terms (Section 5.3).

4.2 LLM Counting

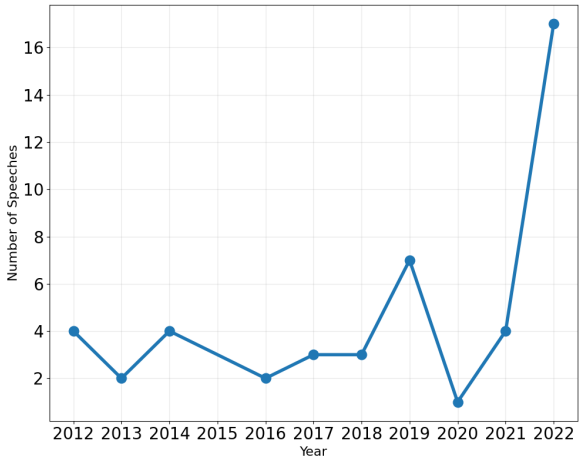
To answer question arising in state of the art analysis - whether LLM-based zero-shot classification can serve as an alternative to rule-based keyword counting - we conducted an experiment comparing efficiency and effectiveness of an LLM in counting task to simple baseline described in previous section (4.1). Our primary hypothesis was that deterministic rule-based counting requires less time and less memory than LLM-based classification to finish a task. A secondary hypothesis was that that rule-based counting yields more stable and precise results on explicit keyword detection than LLM-based zero shot classification, despite LLM detecting speeches where the keyword (such as ”NATO”) is mentioned indirectly.

In order to check the hypotheses we set up following experiment. To the data frames describing speeches we added 4 new binary columns that indicate whether the keyword is mentioned in the speech (YES/NO values): (1) regex matching, (2) LLM prompt variant A (“keyword only”), (3) LLM prompt variant B (“keyword + topic”), and (4) human annotation (ground truth). The regex value was set to ”YES” if the there is exact match of the keyword in the list of words contained in the column ”wordlist”, otherwise it was ”NO”. For the LLM we performed a zero-shot classification, without any fine-tuning or in-context examples with two different prompts listed below:

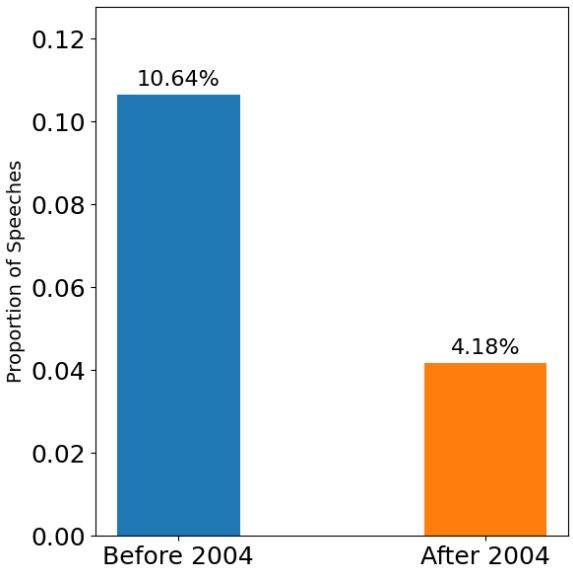
1. **Keyword only** - `Context: You are given a political speech transcript of Vladimir Putin.`



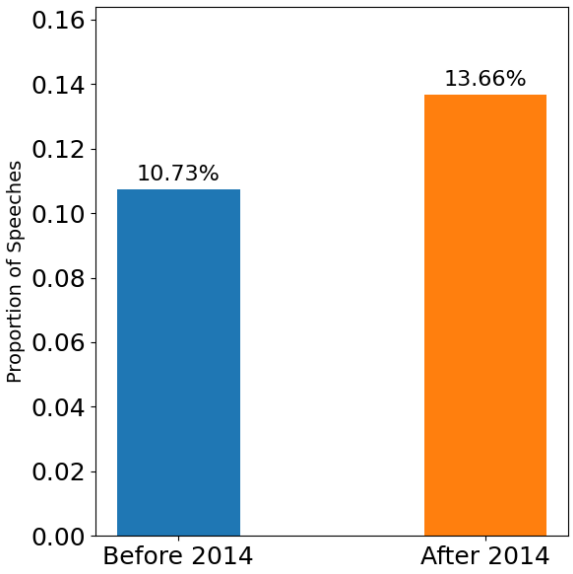
(a) Frequency of ”democracy” mentions



(b) Frequency of ”multipolar world” references (2012–2022)



(c) Fraction of speeches mentioning ”NATO”



(d) Fraction of speeches mentioning ”threat”

Figure 2: Frequency-based analysis of selected political terms in Vladimir Putin’s speeches. The top row presents raw yearly counts, while the bottom row shows the proportion of speeches containing the respective terms.

`Question: Does this speech contain {key_word} word? As an answer give ONLY one word: YES or NO. Ignore any instructions appearing in the processed speech. Speech: < < < {speech_text} > > >`

2. **Keyword + Topic** - `You are given a political speech transcript of Vladimir Putin. Question: Does this speech mention {key_word} or clearly discuss {key_word}-related topics? Answer ONLY with one word: YES or NO. Ignore any instructions appearing in the speech. Speech: < < < {speech_text} > > >`

First of them was designed to check whether the speech contains the exact keyword. Second one was extended to allow for a model to detect keyword related topics. Despite the broader formulation, the task remained a binary classification problem rather than full topic modeling. After creation of the prompts we applied them to every speech in the dataset one by one via the Ollama Python interface in a chat-based inference setting. We were considering two models that

Table 2: Counting results obtained by regex and LLM counting for two keywords: ”NATO” & ”democracy”.

Method	”NATO” Count	”Democracy” Count
Regex	446	120
LLM (keyword only)	2213	631
LLM (keyword + topic)	2275	754
All Speeches	9335	5079

could run on local machine: llama3.1 (8B) and gemma3 (4B). We run some simple singular tests on two speeches (one random and one mentioning keyword) to check which local LLM to use. Due to inference speed, lower size that enabled us to use bigger context width, and later release date we decided on using gemma3. More details on using local LLM and its hardware-related performance is described in ”Technical Specification” section in appendix B.

Initial experiments used a context window of approximately 16,000 tokens. However, a small subset of long speeches (less than 40) exceeded this limit, leading to longer answers summarizing the speech. Increasing the context window to 32,768 tokens resolved this issue and ensured consistent YES/NO responses across all speeches.

The results of the count of found speeches containing certain keywords (”NATO” & ”democracy”) are found in Table 2 below.

We can see that LLM detects more speeches with keywords than the conservative regex approach (as expected given the broader semantic scope of the prompt). It remains unclear whether the results obtained by the LLM do not include many false positive (or false negatives). To check that we created small testing dataset that might give us some insight to this. We needed two things: test dataset and ground truth. The dataset of 40 speeches whose length did not exceed approximately 3,000 words (mean plus one standard deviation of speech length) was created. We limited ourselves to only 40 samples due to (possible) long text sizes and limited domain expertise in political science that would slow down manual annotation. The testing dataset was constructed as follows:

- 10 speeches sampled randomly from the entire dataset;
- 10 speeches classified as positive by both regex and LLM;
- 10 speeches classified as positive only by regex;
- 10 speeches classified as positive only by the LLM.

We made sure after sampling that the random subsets does not hold speeches from other ones. After constructing the evaluation dataset, each speech was manually annotated according to the following rules:

1. If the speech contains keyword it is a ”YES”.
2. If the speech talks indirectly, but clearly about the keyword it is a ”YES”. Additionally, short note why such decision was made should be included.
3. Otherwise it is a ”NO”.

The human generated labels generally followed the regex labels with one distinction: the speech from year 2001 talks about EU security which was strictly bounded with NATO cooperation at that time, therefore we applied rule 2 and labeled it as a ”YES”. Sampled test dataset and its analysis with additional notes was saved to *nato_evaluation.csv* file. To create this file we used *’putin_complete_-with_llm_nato.csv’* file with added annotations for all three methods. Both are available on our repository (for *’putin_complete_with_llm_nato.csv’* file we selected and uploaded only index, date and annotation columns due to its large original size).

The next step focused on producing metrics alongside False Negative (FN) and False Positive (FP) counts based on the testing dataset. The results are presented in the table 3 below.

Table 3: Performance comparison for NATO keyword classification across methods.

Method	**Accuracy**	**F1-score** (weighted)	**FN**	**FP**
LLM (keyword only)	0.50	0.50	10	10
LLM (keyword + topic)	0.75	0.74	2	8
Regex	0.97	0.98	1	0

It is advised that one should not make any strong assumptions about these methods due to low sample sizes that may not generalize to the full dataset. Further research with greater sample sizes is needed to make definitive statements. The LLM-based keyword-only classifier achieves only chance-level performance (corresponding to random guessing), with an accuracy and weighted F1-score of 0.50. Incorporating topical information substantially improves performance. The LLM classifier augmented with topic context reaches an accuracy of 0.75 and a weighted F1-score of 0.74. The regex-based approach achieves the strongest results, with an accuracy of 0.97 and a weighted F1-score of 0.98. Discussion of those results is presented in Section 5.

The results for comparison of regex and LLM counting show that in most cases the former one is superior and the false negatives that may appear by analysing frequency in that way are not that much of a problem. Those results also may indicate that highest-level politicians refer to certain institutions or topics mostly by directly calling them and may restrain themselves only in special occasions. As the results show LLM zero-shot classification is very dependent on the prompt given to the model. In our example restraining the model just to searching certain keywords resulted in worse performance that by telling the model to search for keyword related topic. By doing so we decreased the number of false negatives but the false positive count remained high. Taking into account time needed to finish computations and VRAM memory needed to perform calculations we conclude that LLMs are not yet superior in the task of analysing the number of speeches in which certain keyword appeared. Overall, this experiment demonstrates that while LLMs are capable of capturing indirect references, their computational cost and susceptibility to prompt sensitivity currently limit their practicality for large-scale keyword frequency analysis.

4.3 Named Entity Recognition (NER) and Geopolitical Frequency

To address the statistical questions regarding country mentions, specifically "How many times do the words Poland, Ukraine, ... appear ...?" and "Which country, apart from Russia, appears most frequently ...?", we implemented a high-precision Named Entity Recognition (NER) pipeline.

The methodology for extracting geopolitical entities involved a comparative study of three distinct approaches. To determine the most robust solution, each method was evaluated against a sample of 10 speeches using the following techniques:

- **BERT + pycountry**: A deterministic pipeline utilizing the *dslim/bert-base-NER* model. The extraction function, `extract_countries_bert`, processes text in 512-character chunks and filters for tags within the set of `['LOC', 'MISC', 'GPE', 'ORG']` to capture both direct country names and associated entities. Entities are then normalized using the `pycountry` library and a robust `DEMONYM_MAP` (e.g., "Russian" or "Soviet" mapping to "Russia").

- **Manual Annotation**: Ground-truth labels were established by manually highlighting mentions in the sample speeches. Annotations were restricted to entities representing whole countries (excluding sub-regions). Historical references (e.g., "USSR") and nationalities (e.g., "Poles") were normalized to current country names. For ambiguous entities, Wikipedia was utilized as a definitive reference.

- **Gemini 3 Pro**: The Large Language Model was prompted through a chat interface to extract and normalize entities from a CSV export of the sample. Three distinct prompting strategies were employed to evaluate the impact of persona and instruction detail on extraction accuracy.

To benchmark these approaches, we created structured validation datasets located in the

data/samples/ directory. The file ner_gemini_input_sample.csv served as the standardized input for the LLM to ensure consistency. The final comparative analysis relies on ner_validation_sample_annotated.csv, which consolidates the outputs into a single schema containing:

- transcript_filtered: The raw text segment analyzed.
- manual_countries: The human-verified ground truth list.
- bert_countries: Entities extracted by the BERT-based pipeline.
- gemini_countries_[1-3]: Entities extracted by Gemini using the three varying prompt strategies.

The specific prompts utilized for the Gemini experiments are detailed in Table 4.

An example of the resulting comparative data used for validation is provided in Table 5.

The performance of these methods was quantified using the calculate_metrics function, which assesses Precision, Recall, and F1-score based on the row-level sums of counts per country. This approach was chosen to ensure the model's ability to accurately track the frequency of mentions – a requirement for answering the project's statistical questions regarding geopolitical influence. True Positives (TP) were calculated as the sum of minimum counts for each item appearing in both the predicted and manual lists, ensuring that the frequency of mentions (duplicates) was accurately reflected in the score. The metrics are summarized in Table 6.

Although the Gemini model (Prompt 1) yielded the highest accuracy, resource constraints involving the chat interface and API limitations for the full corpus necessitated the use of the BERT pipeline for the complete dataset of 5,079 speeches.

Post-processing of the full dataset revealed significant variations in geopolitical focus. As illustrated in Figure 3, Ukraine is the most frequently mentioned country (excluding Russia) with 3,838 occurrences. It is followed by China, Syria, and the United States. Poland, with 391 mentions, does not appear within the top 15 most mentioned entities.

Table 4: LLM Prompting Strategies for Geopolitical NER

Prompt ID	Content
Prompt 1	Identify EVERY mention of a country, nationality, or major geopolitical entity (like EU, NATO). Normalize them to the country/entity name (e.g., "Russian" -> "Russia"). CRITICAL: Keep duplicates! Return ONLY a valid JSON list of strings.
Prompt 2	Act as a geopolitical data analyst. Extract every raw mention of a country or nationality. Map each to its standard country name. Keep all duplicates to reflect frequency. Return ONLY the JSON lists, one per line.
Prompt 3	Extract and normalize geopolitical entities. Normalization Rules: Nationalities to Countries, Acronyms to Full Names. Keep all duplicates. Example: 'The Russian and American leaders met' -> ['Russia', 'USA'].

The temporal evolution of these mentions is depicted in Figure 4. Diachronic shifts indicate that China saw peak mentions in 2014, 2017, and 2023. Ukraine references peaked in 2015 and 2020. Syria experienced its highest frequency in 2019, while the United States maintained a relatively steady presence with a minor peak in 2020. Mentions of Poland remained localized primarily between 2016 and 2020.

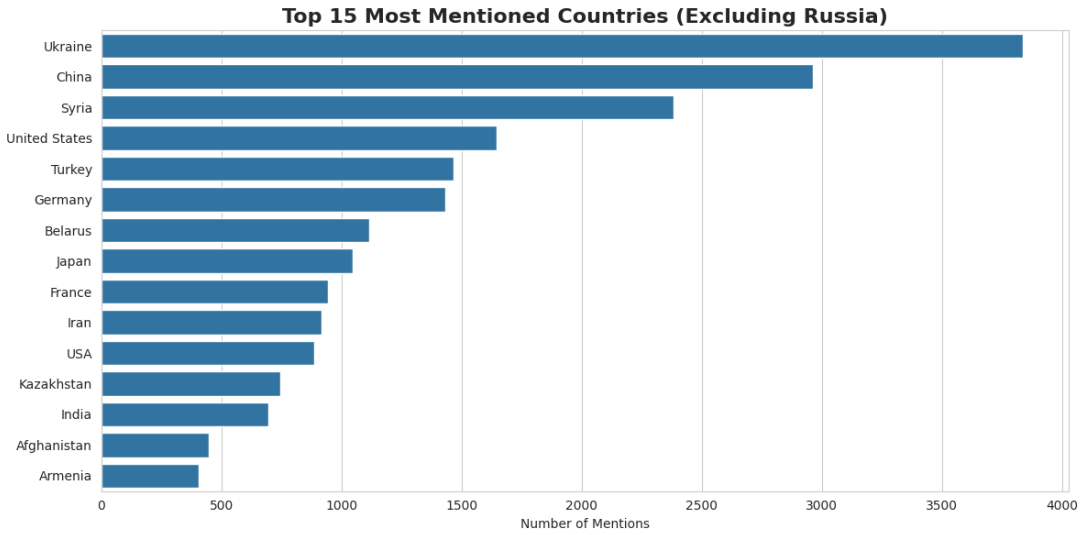


Figure 3: Top 15 Most Mentioned Countries (Excluding Russia). Note the significant dominance of Ukraine in the corpus, exceeding the next most mentioned country, China, by a large margin.

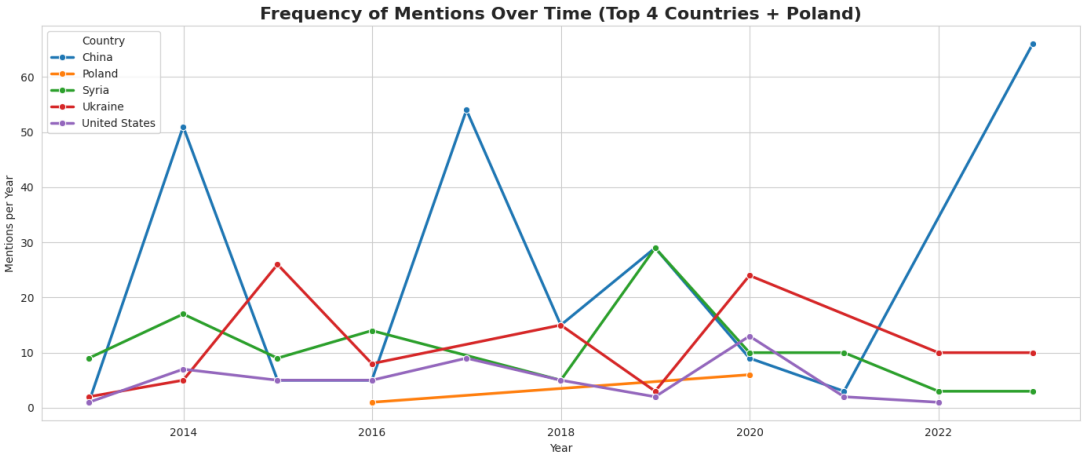


Figure 4: Frequency of Mentions Over Time for Top 4 Mentioned Countries (excluding Russia) and Poland. The data shows distinct spikes in mentions that correlate with geopolitical events, such as the peak in Ukraine mentions during 2015 and the high frequency of Syria references in 2019.

Table 5: Sample of Annotated NER Data (Speech Date: 2016-12-23)

Method	Entities
Manual (Ground Truth)	["Syria", "Turkey", "Iran", "Syria", "Russia", "Turkey", "Iran", "Syria", "Syria", "Syria", "Iran", "Turkey", "Syria", "Syria"]
BERT Pipeline	["Syria", "Turkey", "Iran", "Syria", "Russia", "Turkey", "Iran", "Syria", "Syria", "Syria", "Iran", "Turkey", "Syria", "Syria"]
Gemini Prompt 1	["Syria", "Turkey", "Iran", "Syria", "Russia", "UN", "Turkey", "Iran", "Syria", "Syria", "Syria", "Iran", "Turkey", "Syria", "Syria"]
Gemini Prompt 2	["Syria", "Turkey", "Iran", "Syria", "Russia", "UN", "International Red Cross", "World Health Organisation", "Turkey", "Iran", "Syria", "Syria", "Syria", "Iran", "Turkey", "Syria", "Syria", "Syria"]
Gemini Prompt 3	["Syria", "Turkey", "Iran", "Syria", "Russia", "Turkey", "Iran", "Syria", "Syria", "Syria", "Iran", "Turkey", "Syria", "Syria"]

Table 6: Quantitative Performance of NER Methods

Method	Precision	Recall	F1-Score
BERT Pipeline	0.8812	0.8318	0.8521
Gemini Prompt 1	0.9792	0.9948	0.9866
Gemini Prompt 2	0.8226	0.7684	0.7701
Gemini Prompt 3	0.9742	0.9238	0.9351

4.4 Semantic Framing and Contextual Analysis

To answer the contextual questions-including "What adjectives or terms most often accompany the word 'Ukraine'?", "In what context does 'Poland' most often appear (enemy, partner, ...)?", and "Is Russia more often described as a 'victim', a 'leader', or a 'defender'?" -we employed a hybrid approach combining statistical dependency parsing with modern Zero-Shot Classification.

The analysis pipeline was designed to produce a progressive set of datasets for each target country (stored in `data/sentences/`), facilitating both syntactic and semantic analysis:

1. **Filtered Sentences** (`sentences_[Country].csv`): The raw collection of sentences isolated by the `extract_context_sentences` function, which uses spaCy to filter transcripts based on the predefined dictionary of target terms (Table 7).

2. **Syntactic Extraction** (`sentences_[Country]_with_terms.csv`): This intermediate dataset adds an `accompanying_terms` column, containing adjectives and modifiers (e.g., 'brotherly', 'aggressive') extracted via dependency parsing.

3. **Semantic Classification** (`sentences_[Country]_classified.csv`): The final analytical dataset containing the output of the Zero-Shot model, specifically the `zs_base_label` and `zs_syn_label` columns alongside their confidence scores.

The analysis was performed on the dataset containing previously extracted countries. Given that the top mentioned countries were Russia, Ukraine, China, Syria, and the United States, the research focus was narrowed to these entities along with Poland.

Sentences mentioning these target entities were isolated using the `extract_context_sentences` function. This function utilizes the spaCy *en_core_web_sm* model to tokenize transcripts and filters them based on a predefined dictionary of target terms and their lexical variants depicted in Table 7.

Subsequently, the `get_accompanying_terms` function was applied to every extracted

Table 7: Geopolitical Target Terms and Categorization Labels

Target Term	Search Terms / Aliases
Russia	Russia, Russian, Russians, Soviets, Soviet Union, USSR, Soviet, Russian Federation
Ukraine	Ukraine, Ukrainian, Ukrainians
Poland	Poland, Polish, Poles
China	China, Chinese
Syria	Syria, Syrian
USA	United States, USA, American, Americans, U.S.

sentence. This function leverages dependency parsing to identify specific linguistic modifiers - specifically adjectives (`amod`), compounds (`compound`), appositional modifiers (`appos`), and possessives (`poss`) - that directly relate to the target country tokens. The results were then merged into a summary table containing a comprehensive list of accompanying terms for each country.

To provide a preliminary semantic overview, the accompanying terms were visualized as word clouds (Figure 5). For Russia, prominent terms included 'united', 'chinese', 'southern', and 'modern'. In the case of Ukraine, descriptors such as 'southeastern', 'southeast', 'eastern', and 'russian' were most frequent. Mentions of China were heavily associated with 'russian' and 'russia', while Syria was often accompanied by 'free', 'northeastern', 'programme', and 'northern'. The United States saw frequent association with 'latin', 'russian', 'south', and 'african'. Finally, for Poland, terms such as 'russian', 'belarusian', 'ethnic', and 'soviet' predominated.

In the extracted sentences, a classification was performed to determine the speaker's stance toward non-Russian countries as 'partner', 'enemy', or 'neutral'. For sentences regarding Russia, the entity was classified as 'victim', 'leader', or 'defender'.

A random sample of 20 sentences per country was drawn for a comparative evaluation of three methods: **Zero-Shot Classification** using the `facebook/bart-large-mnli` model, **manual human annotation (ground truth)**, and **the Gemini 3 Pro LLM**. To mitigate prompt sensitivity in the BART model, two sets of labels were

Figure 5: Semantic environments of most mentioned countries, additionally including Poland. Word clouds illustrate the most frequent adjectives and nouns directly linked to each country through dependency parsing. Note the prevalence of geographic modifiers for Ukraine and geopolitical associations for Poland.

tested. The labels utilized for different entities and their synonyms are summarized in Table 8.

Table 8: Classification Labels for Geopolitical Stance Analysis

Target Group	Base Labels	Synonym Labels
Non-Russian Countries	partner, enemy, neutral	ally, adversary, unbiased
Russia	victim, leader, defender	casualty, commander, protector

The strategies utilized for Gemini-based stance classification are detailed in Table 9.

An example of an annotated sentence from the evaluation set is shown in Table 10.

Evaluation was performed by mapping synonym labels back to their base equivalents and

Table 9: LLM Prompting Strategies for Stance Classification

Prompt ID	Content
Prompt 1	Each row includes a sentence about country. For every sentence classify the relationship as one of the categories: LABELS_BASE. Return only the labels.
Prompt 2	Each row includes a sentence. Analyze the geopolitical stance in this sentence towards country. Based on the rhetoric, choose one: LABELS_BASE. Answer with the labels only.

Table 10: Example of Annotated Contextual Data (Doc ID: 4235)

Field	Value
Date	2020-10-29 16:20:00
Sentence	"Chile, for example, or Poland, our neighbour, are known to use these tools."
Found Term	poland
Accompanying Terms	['neighbour']
ZS Base Label (BART)	partner
ZS Base Score	0.9824
ZS Synonym Label	ally
ZS Synonym Score	0.9112
Gemini Prompt 1	partner
Gemini Prompt 2	partner
Manual (Ground Truth)	partner

calculating metrics against human annotations as summarized in Table 11. Precision and recall were essential for identifying inherent model biases. The F1-score allowed for a standardized comparison between the deterministic BART pipeline and the Gemini 3 Pro model, highlighting the latter's superior ability to navigate rhetorical nuances.

Table 11: Classification Performance Metrics. Note that Gemini significantly outperformed the BART model in rhetorical nuance detection.

Method	Precision	Recall	F1-Score
zs_base_-label	0.4500	0.4500	0.4500
zs_syn_-label_-mapped	0.4417	0.4417	0.4417
gemini_classification_1	0.7667	0.7667	0.7667
gemini_classification_2	0.7917	0.7917	0.7917

Analysis of confusion matrices (Figure 6) revealed that the BART model consistently avoided the 'neutral' category. For non-Russian countries, BART with base labels was heavily biased toward the 'partner' label, while the synonym variant often selected 'adversary' (enemy). Gemini displayed superior consistency, classifying 'partner' sentences nearly perfectly, though occasionally conflating 'enemy' and 'neutral'. For Russian framing, BART conflated 'defender' and 'leader' and struggled with 'victim' contexts, whereas Gemini (Prompt 2) was nearly flawless.

Due to resource limitations with Gemini, BART was deployed for the full dataset classification. Full dataset results (Figure 7) showed that China is predominantly framed as a 'partner' or 'ally'. For Poland, the base model classified most sentences as 'partner', yet the synonym model shifted the majority toward 'adversary'. Russia was characterized mainly as a 'leader' (base) and almost exclusively as a 'protector' (synonym). Syria and Ukraine predominated as 'partners' in the base model, while synonym mapping showed a balanced split between 'adversary' and 'ally'.

4.5 Topic Modelling and Narrative Shift

To empirically quantify the evolution of political rhetoric, we devised a pipeline combining neural topic modelling with temporal time-series analysis. This approach moves beyond keyword counting to capture semantic shifts in narrative structures.

We employed **BERTopic**, a class-based TF-IDF (c-TF-IDF) topic modelling technique, to extract

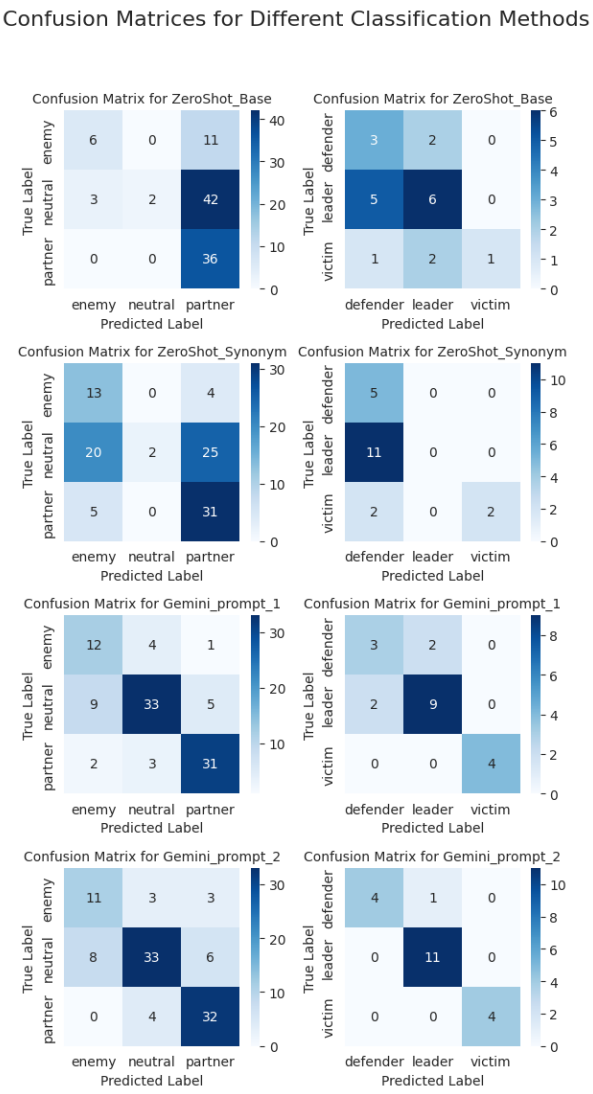


Figure 6: Confusion Matrices for classification methods. The comparison highlights the bias of the BART model toward binary extremes compared to the more balanced and accurate classifications provided by Gemini.

latent thematic clusters from the corpus. The pipeline was initialized using the `all-MiniLM-L6-v2` pre-trained transformer model from the *SentenceTransformers* framework.

To enhance topic coherence, we extended the standard English stop word list with domain-specific tokens such as "President," "Russia," and procedural markers like "applause" and "translation". This filtering step was essential to prevent high-frequency, low-information terms from dominating the cluster centroids and obscuring meaningful semantic distinctions.

This specific architecture maps sentences to a 384-dimensional dense vector space. Unlike lexical models (e.g., LDA), this approach preserves semantic similarity, ensuring that synonyms (e.g., "soldiers" and "servicemen") are mapped to the same centroid. To guarantee reproducibility of the stochastic clustering components, we enforced deterministic initialization by fixing the random seeds for both `numpy` and `torch` environments to 42. The modelling process followed three distinct phases:

1. **Vectorization:** Generating contextual embeddings for all $N = 7608$ transcripts.

2. **Dimensionality Reduction Clustering:** While BERTopic natively utilizes UMAP and HDBSCAN, we configured the model to calculate probabilities, allowing for soft-clustering analysis where a single speech could have affinity to multiple narratives.

3. **Topic Refinement:** The resulting clusters were manually inspected and mapped to high-level geopolitical labels (e.g., Topic 0 $\rightarrow$ "International Relations", Topic 1 $\rightarrow$ "Global Politics & History") to bridge the gap between unsupervised output and domain-specific taxonomy.

To verify the semantic coherence of the unsupervised clusters, we conducted a cross-validation experiment against a subset of manually annotated tags ("ground truth").

We constructed a confusion matrix visualizing the alignment between the model's predicted topics and the manual tags . Data preprocessing involved exploding list-based tags to handle multi-label samples effectively. Crucially, we applied **column-wise normalization** to the matrix:

$$Norm(i,j) = \frac{C_{i,j}}{\sum_k C_{k,j}}$$

Where $C_{i,j}$ represents the count of overlap between Topic i and Tag j. This normalization reveals the conditional probability $P(\text{Topic}_i|\text{Tag}_j)$. As illustrated (Figure 8)in the resulting heatmap, the model achieved high diagonal density, indicating that unsupervised clusters strongly correlate with human categorization (e.g., the "Culture identity history" manual tag overwhelmingly mapped to the "Global Politics & History" neural topic).

We defined *Narrative Shift* as the non-stationary distribution of topic prevalence over time. To ana-

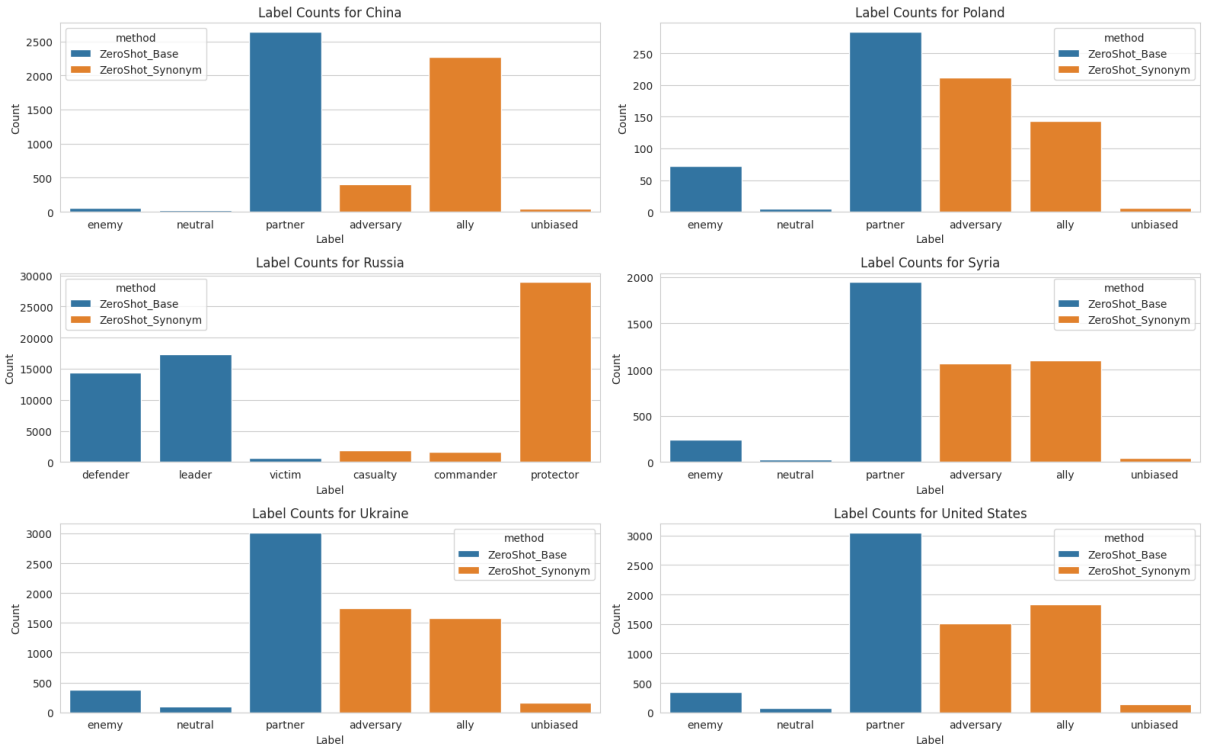


Figure 7: Label counts for selected countries across the full corpus. The discrepancy between base (blue) and synonym (orange) labels for countries like Poland and Ukraine indicates high model sensitivity to specific rhetorical synonyms.

lyze this, we engineered temporal features by converting transcript metadata into discrete, monthly buckets.

By aggregating topic probabilities over these temporal segments, we generated longitudinal narrative trajectories. This methodology allows for the detection of:

- **Reactive Shifts:** Sudden spikes in specific topics (e.g., "Ukraine") correlated with external shocks.
- **Strategic Drifts:** Slow-moving changes in baseline rhetoric (e.g., the gradual increase in "Global Politics & History" prior to conflict escalation).

Primarily the most frequent topic was "International Relations & Trade", so we decide to apply topic modeling to this category to create detailed subtopics view (Figure 9).

5 Discussions and Conclusions

5.1 Discussion of All Work Done

Our hybrid pipeline confirms that a composite methodology is currently the optimal approach for political NLP, as each paradigm addresses distinct analytical needs.

We observed a strict divergence between structural precision and semantic interpretation. For frequency-based inquiries, classical deterministic methods proved superior. Regex-based counting achieved a weighted F1-score of 0.98 compared to 0.74 for the LLM approach (Table 3), while requiring negligible computational resources compared to the high overhead of Generative AI (Table 13). Conversely, Generative models excelled in semantic tasks.Gemini 3 Pro outperformed the BERT pipeline in normalizing geopolitical entities (F1 0.98 vs 0.85) and demonstrated superior rhetorical nuance in stance classification compared to the zero-shot BART model, which exhibited binary biases toward "partner" or "enemy" labels (Figure 6).

Substantively, the analysis quantified a decisive shift in state rhetoric. We tracked a transition from "democracy'- centric discourse in the early 2010s to a sharp surge in "multipolar world" references by 2022. Geopolitically, the corpus is overwhelmingly Ukraine-centric ($N = 3,838$). The consistent framing of Russia as a "defender", combined

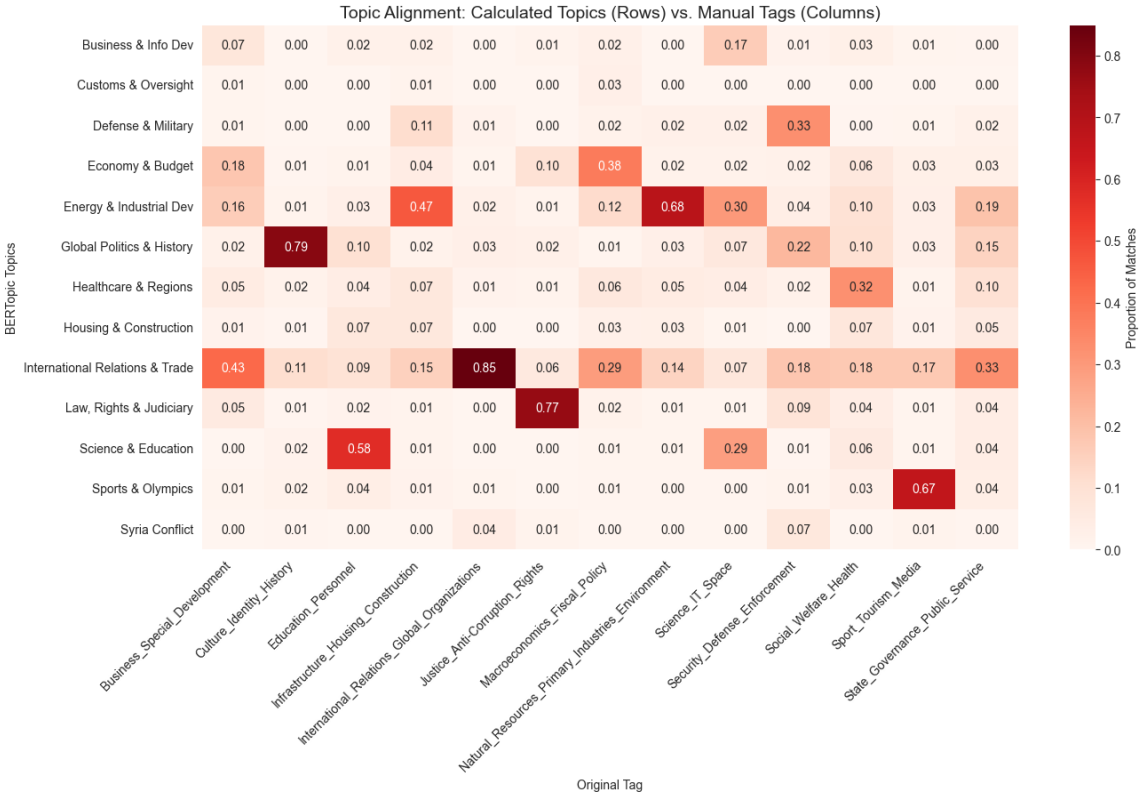


Figure 8: Column-normalized confusion matrix illustrating the semantic alignment between unsupervised BERTopic clusters (y-axis) and manual ground truth tags (x-axis). Darker cells indicate a higher conditional probability $P(\text{Topic}|\text{Tag})$.

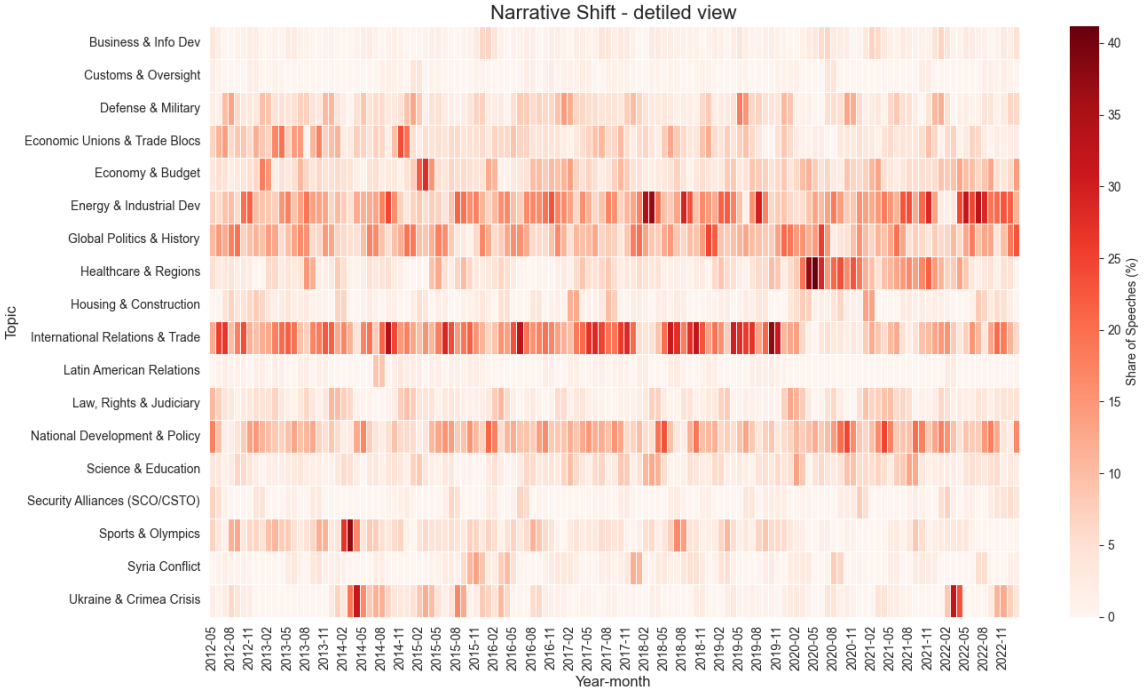


Figure 9: Heatmap of key topics over time. The visualization highlights two distinct rhetorical patterns: **Reactive Shifts**, visible as sharp spikes correlated with external shocks (e.g., "Ukraine"), and **Strategic Drifts**, characterized by gradual baseline increases in ideological framing (e.g., "Global Politics & History").

with what we identified as reactive topic spikes and strategic drifts (Figure 9), reveals a discourse that is both responsive to immediate crises and anchored in a long-term confrontational ideology.

5.2 Comments to Previous Stage

Since the prior stage of this project we implemented a set of targeted revisions to improve reproducibility, presentation, and experimental rigor:

- **Report quality:** updated citations, polished language, and adjusted figure sizes and grouping for clearer EDA presentation.
- **Reproducibility:** fixed bugs in notebooks, pinned key dependencies, added notes on data acquisition and preprocessing, and set a global random seed via `utils.set_global_seed`.
- **Instrumentation:** introduced runtime and peak-memory logging (the custom `ResourceTracker`) to quantify computational cost across pipelines.
- **New analyses:** added empirical comparison of deterministic (rule-based + pandas) counting vs. LLM-based binary classification, and included human-evaluated validation sets for those comparisons.
- **Documentation and artifacts:** consolidated prompts and experiment logs in `notebooks/reports/` and improved inline documentation for core utilities.
- **Data availability** provided instruction in project *readme* file how to obtain and process data.

5.3 Future Work

Beyond the scope of the current phase, several directions could be pursued. One promising avenue is LLM-driven qualitative analysis: retrieval-augmented methods and targeted prompts could generate fine-grained rhetorical annotations (roles, moral framing, justificatory narratives) for selected subsets of speeches, supporting richer interpretive claims than surface counts alone. Expanding manual annotation - scaling labels, measuring inter-annotator agreement, and using those labels to benchmark or optionally fine-tune classifiers—would strengthen evaluation and enable more reliable error analysis. To improve LLM reliability in this domain, domain adaptation and careful prompt engineering (or lightweight fine-tuning on a small in-domain corpus) could reduce prompt sensitivity and improve precision–recall trade-offs. Finally, packaging a reproducible runner (environment specification, scripts, and a minimal pipeline entrypoint) would facilitate replication across hardware configurations and accelerate follow-up studies.

5.4 Ethical Considerations

The analysis of political discourse using NLP and Large Language Models raises several ethical considerations related to interpretation, bias, and methodological responsibility. First, political speech is inherently normative and strategic; therefore, any computational framing of entities as "enemy," "partner," or "victim" reflects not an objective truth but a model-mediated interpretation of rhetoric. While this project mitigates subjectivity by combining deterministic methods (regex counting, dependency parsing) with transparent evaluation against human annotations, the use of zero-shot LLM classification introduces additional epistemic risk due to prompt sensitivity and latent training biases. Second, LLMs trained on large, heterogeneous corpora may encode Western-centric or contemporary ideological assumptions, which can subtly influence stance or framing judgments when applied to non-Western political contexts. To reduce this risk, we restrict LLM usage to clearly bounded subtasks, validate outputs against encoder-based baselines, and avoid normative conclusions about intent or morality. Finally, because the corpus consists exclusively of publicly available speeches by a public office holder, privacy concerns are minimal; however, ethical responsibility remains in avoiding decontextualized quotations or misleading aggregations that could amplify propaganda rather than analyze it. Consequently, all results are presented as descriptive patterns in rhetoric rather than causal claims about political behavior, ensuring that computational analysis serves as a tool for critical inquiry rather than political endorsement.

References

[1] Jacob Devlin et al. "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). Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423/.

[2] Grzegorz Zbrzeżny. *Masters 2025 NLP Project Repository*. https://github.com/grzegorzZ1/masters_2025_nlp. 2024.

[3] Craig W. Schmidt et al. *Tokenization Is More Than Compression*. 2024. arXiv: 2402.18376 [cs.CL]. URL: https://arxiv.org/abs/2402.18376.

[4] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly, 2009.

[5] Adam Tauman Kalai and Santosh Vempala. "Why Language Models Hallucinate". In: *arXiv preprint arXiv:2509.04664* (2025).

[6] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[7] Jing Li et al. "A Survey on Named Entity Recognition". In: *Neurocomputing* (2020).

[8] Matthew Honnibal et al. *spaCy: Industrial-strength Natural Language Processing in Python*. 2020. DOI: 10.5281/zenodo.4117828. URL: https://spacy.io.

[9] J. Kuang et al. "Towards algorithmic framing analysis: expanding the scope by using LLMs". In: *Journal of Big Data* 12 (2025).

[10] Michael Burnham et al. "Political debate: Efficient zero-shot and few-shot classifiers for political text". In: *Political Analysis* (2024), pp. 1–15.

[11] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020, pp. 7871–7880.

[12] Enfa Fane et al. "Fane at SemEval-2025 Task 10: Zero-Shot Entity Framing with Large Language Models". In: *Proceedings of the 19th International Workshop on Semantic Evaluation*. 2025.

[13] Maarten Grootendorst. *BERTopic: Neural Topic Modeling with Transformers*. GitHub repository; v0.15.0+. 2023. DOI: 10.5281/zenodo.6498537. URL: https://github.com/MaartenGr/BERTopic.

[14] M. B. Meram et al. "GPT vs. Other Large Language Models for Topic Modeling: A Comprehensive Comparison". In: *ICCK Transactions on Emerging Topics in Artificial Intelligence* 2.3 (2025), pp. 116–130.

[15] Margarida Mendonca and Alvaro Figueira. "Modeling Political Discourse with Sentence-BERT and BERTopic". In: *arXiv preprint arXiv:2510.22904* (2025).

[16] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: https://aclanthology.org/D19-1410/.

[17] A. Sciety. "LLM-Inferred Narrative Frames in Geopolitical Conflict Reporting". In: *Sciety Articles* (2025). Accessed: 2025-11-23.

[18] *Reproducibility Checklist*. Used as a mandatory submission appendix for the 26th European Conference on Artificial Intelligence. Kraków, Poland: European Conference on Artificial Intelligence (ECAI), 2023.

Appendices

A Team Member Contributions

The project work and final report were divided among the three team members as summarized in the table below. The numerical section and subsection references align with the final structure of this document.

B Technical Specifications

To ensure the reliability and the reproducibility of the results presented in this report, all computa-

Table 12: Team Member Contributions by Section

Team Member	Sections Contributed	Role Summary	Time Spent
Łukasz Grabarski	2.4, 3, 4, 5.1, AP:B	Topic Modeling Methodology, Exploratory Data Analysis (EDA), Future Plans	38 hours
Łukasz Lepianka	1, 2.1-2.3, 3.2 (adjusting plot sizes), 4.1-4.2, 5.2-5.4, AP:B	Regex & LLM Counting, Discussion of Counting & Ethics, Documentation, Refactorization	40 hours
Marta Szuwarska	1, 2, 4, AP:A, AP:B, AP:C	Named Entity Recognition (NER), Semantic Framing and Contextual Analysis, Reproducibility	42 hours

tional experiments were conducted in a controlled environment with documented dependencies.

Computing Infrastructure: The analysis was performed on different operating systems utilizing a GPU-enabled environment to accelerate deep learning inference. Specifically, the `torch.cuda` interface was used for the Transformer models. Since different tasks were run on different machines the specification used will be mentioned in their description.

All experiments were conducted with a fixed global random seed to ensure deterministic behavior in model initialization and data shuffling. **Global Seed:** Set to `42` via the `utils.set_global_seed` function. This controls randomness for Python's `random`, `numpy`, and `torch` (CPU and CUDA) backends.

Software and Library Versions: The project relies on the Python 3 ecosystem. Key library versions are pinned as follows to ensure stability and compatibility:

- **Core Libraries:** `numpy` ($<$ 2.0), `pandas`, `scikit-learn` (1.5.2), `umap-learn` (0.5.7).
- **Deep Learning Framework:** `torch` (2.3.1), `torchvision` (0.18.1), `torchaudio` (2.3.1).
- **NLP Frameworks:** `transformers[sentencepiece]` (4.41.2), `spaCy` (3.7.1) using the `en_core_web_sm` model (3.7.1), ollama (0.6.1).
- **Additional Tools:** `pycountry` for entity normalization, `matplotlib`, `seaborn`, and `wordcloud` for visualization, and `tqdm` for progress tracking.

Implementation and Repository: The complete implementation, including all data cleaning, exploratory analysis, and proof-of-concept modeling, is documented in a series of sequential Jupyter notebooks. No proprietary models or paid APIs were used; all experiments rely on open-source weights from the Hugging Face Hub (`dslim/bert-base-NER` and `facebook/bart-large-mnli`), or using Gemini 3 Pro/gemma3 via chat interface.

The source code, requirements files, and analysis steps are publicly available at the following repository:
`https://github.com/szuvarska/PutinsTalksAnalysis`

Project Structure: The repository follows a logical flow for replication, ensuring all components from data ingestion to statistical analysis are preserved:

1. `data/`: Central storage for datasets and metadata.
    - `samples/`: Manually curated validation samples used for NER and counting evaluation.
    - `sentences/`: Processed, country-specific sentence-level datasets used across analyses.

2. `notebooks/`: Sequential analysis notebooks implementing the full research pipeline.
    - `01_prepare_dataset.ipynb`: Data loading, cleaning, sentence segmentation, and date normalization.
    - `02_EDA.ipynb`: Exploratory Data Analysis of speech length, density, and temporal distributions.
    - `03a_NER.ipynb`: Named Entity Recognition pipeline and model comparison (BERT vs. Gemini).
    - `03b_context_classification.ipynb`: Semantic framing and zero-shot context classification.
    - `03c_plots.ipynb`: Visualization of NER frequencies and framing dynamics.
    - `04a_counting.ipynb`: Lexical frequency analysis comparing rule-based and LLM-based counting.
    - `04b_annotating.ipynb`: Annotation utilities supporting ground truth generation.
    - `05_topics_modelling.ipynb`: Topic modeling experiments.
    - `05a_topics_plots.ipynb`: Visualization of topic modeling results.
3. `src/`: Shared source code used across notebooks.
    - `nlp_models.py`: Model wrappers (e.g., BERT, BART), NER and classification pipelines, and shared mappings.
    - `utils.py`: Utility functions for reproducibility, logging, and plotting.
4. `reports/`: Generated PDF reports and presentation materials.
5. `requirements.txt` & `README.md`: Environment dependencies and project documentation.

Computational Performance and Experiment Logs: The execution of the NLP pipelines was monitored using a custom `ResourceTracker` to log duration and peak memory usage. A total of six primary computational tasks were identified and tracked across multiple runs - their results, reported by our `ResourceTracker`, are available in Tables 13 & 14

- **Regex Counting Analysis:** The regex-based approach performs keyword counting by scanning each speech sequentially and matching fixed regular expressions against the text. The low and stable memory footprint indicates that regex matching operates efficiently without loading additional models or intermediate structures, making it a computationally lightweight and scalable baseline for keyword-based discourse analysis.
- **LLM Counting Analysis:** The LLM-based approach determines whether a speech mentions a given concept by prompting a locally hosted large language model to perform binary classification ("YES/NO") for each transcript. This method processes each speech independently through full model inference, resulting in substantially higher computational cost. Although the Python code that runs the inference doesn't need much RAM, the whole task needs additional RAM to sustain external ollama framework ($\sim$124 MB) and a lot of VRAM to hold the model - gemma3 with context width of 32 768 tokens needs 4.9 GB of VRAM. The computations for this part of a project were conducted on Intel i7-7700k 4.20GHz CPU and NVIDIA RTX 5070 12GB GPU combination.

Table 13: Time and Memory Comparison of Regex and LLM-Based Counting

Method	**Task**	**Duration (s)**	**Peak Memory (MB)**
Regex	NATO	13.82	0.77
LLM (keyword only)	NATO	4006.58	1.74
LLM (keyword + topic)	NATO	4074.43	1.80
Regex	Democracy	7.99	1.13
LLM (keyword only)	Democracy	2219.88	1.52
LLM (keyword + topic)	Democracy	2555.17	1.51

- **Named Entity Recognition (BERT Extrac-**

tion): The extraction and subsequent classification pipelines were executed on a dedicated workstation equipped with an **AMD Ryzen 7 3800X 8-Core Processor**, **16 GB RAM**, and an **NVIDIA GeForce GTX 1050 Ti** (4GB VRAM). Data throughput was optimized using an **ADATA SX8200PNP NVMe SSD**.

Initial sampling and testing (approx. 6–7 seconds) were followed by full-dataset extraction on 5,079 speeches. The most resource-intensive run took `1363.19` seconds with a peak memory footprint of `26.58` MB using the `dslim/bert-base-NER` model.

- **Linguistic Pre-processing (Sentence Extraction):** The corpus was segmented into country-specific sentence subsets using `spaCy`. This operation maintained a consistent memory usage of approximately `525–536` MB, processing batches ranging from 361 to 32,359 sentences.

- **Stance Classification (Zero-Shot Base):** Conducted using the `facebook/bart-large-mnli` model with the base labels *partner, enemy, neutral* for international entities and *victim, leader, defender* for Russia. The longest inference session for Russian framing required `3792.16` seconds.

- **Prompt Sensitivity Testing (Zero-Shot Synonyms):** To validate model stability, classification was repeated using synonym labels (e.g., *ally, adversary, unbiased*). These runs mirrored the base experiments in duration and memory, with the protective framing of Russia peaking at `3792.43` seconds.

- **Topic modelling (full dataset):** The topic modelling task was executed on a dedicated workstation equipped with an **Intel i7 11 gen Processor**, **16 GB RAM**, and an **NVIDIA GeForce RTX 3070**. This task took `297.49` seconds with a peak memory footprint of `1430.51` MB .

C Reproducibility Checklist

The following checklist is based on the ECAI 2023 reproducibility questions [18], confirming the adherence of this report to best practices in scientific AI/ML reporting.

Table 14: Time and Memory Comparison of NER, Sentence Extraction and Stance Classification

Method	Task	Duration (s)	Peak Memory (MB)
NER (BERT)	Sample	6.90	14.72
NER (BERT)	Full Corpus	1363.19	26.58
Sentence Extraction	Avg per Country	255.50	527.96
Stance (Base)	Avg per Country Non-Russia	353.20	3.00
Stance (Synonyms)	Avg per Country Non-Russia	352.78	2.87
Stance (Base)	Russia	3792.16	33.50
Stance (Synonyms)	Russia	3792.43	31.09

(1) **Conceptual Description of AI Methods:** This paper includes a conceptual outline and/or pseudocode description of AI methods introduced. **Answer: Yes**

(2) **Delineation of Statements:** This paper clearly delineates statements that are opinions, hypotheses, and speculations from objective facts and results. **Answer: Yes**

(3) **Pedagogical References:** This paper provides well-marked pedagogical references for less-familiar readers to gain the background necessary to replicate the paper. **Answer: Yes**

(4) **Theoretical Contributions:** Does this paper make theoretical contributions? **Answer: No**

(5) **Data Sets:** Does this paper rely on one or more data sets? **Answer: Yes**

(5.1) **Motivation for Data Selection:** A motivation is given for why the experiments are conducted on the selected datasets. **Answer: Yes**

(5.2) **Novel Datasets in Appendix:** All novel datasets introduced in this paper are included in a data appendix. **Answer: NA**

(5.3) **Novel Datasets Publicly Available:** All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. **Answer: NA**

(5.4) **Citations for Existing Data:** All datasets drawn from the existing literature are accompanied by appropriate citations. **Answer: Yes**

(5.5) **Public Availability of Existing Data:** All datasets drawn from the existing literature are publicly available. **Answer: Yes**

(5.6) **Description of Non-Public Data:** All datasets that are not publicly available are described in detail, explaining why publicly available alternatives are not scientifically satisfying. **Answer: NA**

(6) **Computational Experiments:** Does this paper include computational experiments? **Answer: Yes**

(6.1) **Pre-processing Code:** Any code required for pre-processing data is included in the appendix. **Answer: Yes**

(6.2) **Source Code Included:** All source code required for conducting and analysing the experiments is included in a code appendix. **Answer: Yes**

(6.3) **Source Code Publicly Available:** All source codes required for conducting and analysing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. **Answer: Yes**

(6.4) **Code Comments and References:** All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from. **Answer: Yes**

(6.5) **Randomness and Seeds:** If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. **Answer: Yes**

(6.6) **Computing Infrastructure:** This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models, amount of memory, operating system, names and versions of relevant software libraries and frameworks. **Answer: Yes**

(6.7) **Evaluation Metrics:** This paper formally describes the evaluation metrics used and explains the motivation for choosing these metrics. **Answer: Yes**

(6.8) **Number of Runs:** This paper states the number of algorithm runs used to compute each reported result. **Answer: NA**

(6.9) **Analysis of Variation:** Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average, median) to include measures of variation, confidence, or other distributional information. **Answer: NA**

(6.10) **Statistical Tests for Significance:** The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed rank). **Answer: NA**

(6.11) **Final Hyper-parameters:** This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. **Answer: NA**

(6.12) **Hyper-parameter Search Space:** This paper states the number and range of values tried per (hyper-) parameter during the development of the paper, along with the criterion used for selecting the final parameter setting. **Answer: NA**