Description and Methodology of the Automated Political Stance Identification

1 Overview

The **Automated Political Stance Identification (APSI)** framework is a transparent, theory-driven tool for detecting political rhetoric and ideological stance in text. APSI uses **Natural Language Inference (NLI)** to estimate whether a text entails political statements grounded in established political science theory.

Rather than training a supervised classifier on labeled datasets, APSI relies on the `Political DEBATE` (Burnham 2024) pretrained model and applies a **hypothesis-based scoring framework** that evaluates a set of natural-language hypotheses against an input text. This approach enables:

- Zero-shot political stance detection
- Interpretable and auditable outputs
- Minimal training data requirements
- Deterministic scoring and reproducibility
- Scalable analysis of large text corpora

2 Key Design Principles

2.1 Theory-driven classification

Political constructs are derived from established academic definitions and expert survey codebooks, then translated into hypothesis statements suitable for NLI evaluation.

2.2 Transparent scoring

Each stance score can be traced back to:

- Explicit hypotheses
- Entailment probabilities from the NLI model
- Deterministic aggregation rules

2.3 Zero-shot stance detection

By using the `Political DEBATE` model, no additional fine-tuning or labeled training data is required for inference.

2.4 Interpretability

Outputs include a topic relevance pre-check, a numerical score, an interpretation label, confidence estimates, and contradiction detection.

3 Methodology

3.1 Overview

This section shows the methodology behind the outputs produced by the APSI. We provide, as an example, the case of the populist-pluralist rhetoric dimension. But the same methodological approach is used for the other two dimensions already in the tool. The method evaluates how strongly a text supports predefined theoretical statements in the form of hypotheses about each side of the three dimensions.

The tool produces:

- Topic relevance classification
- Score (0–10 scale)
- Confidence estimate
- Contradiction detection

The approach is theory-driven and model-based rather than keyword-based.

3.2 Conceptual Framework

The scoring framework distinguishes two competing stances for each dimension.

- **Economic Left vs Economic Right**
  - Role of government in the economy (e.g., redistribution, taxation, public spending, privatization).
  - Regulation of markets and labor protections (e.g., minimum wage, unions, corporate oversight).
  - Welfare state and social policy (e.g., social safety nets, public services, inequality reduction).
- **Pluralist Rhetoric vs Populist Rhetoric**
  - Framing of political actors (e.g., people vs elites, legitimacy of opposition, anti-establishment claims).
  - Role of democratic institutions (e.g., courts, bureaucracy, checks and balances).
  - Decision-making style (e.g., compromise and negotiation vs direct expression of popular will).
- **Opposition to Liberal Democratic Values vs Support for Liberal Democratic Values**

- Commitment to civil liberties (e.g., freedom of speech, press, and association).
- Support for democratic procedures (e.g., free and fair elections, rule of law).
- Support for institutional constraints on political power.

3.3 Analytical Pipeline

The tool follows five sequential steps:

1. Topic relevance detection
2. Hypothesis entailment scoring
3. Populism–pluralism/Left–Right/Support liberal democracy–Opposition liberal democracy aggregation
4. Confidence estimation
5. Score interpretation

3.4 Topic Relevance Pre-Check

Before scoring, the tool determines whether the input text concerns democratic governance or political legitimacy.

The model evaluates the text against topic hypotheses related to political authority, institutions, decision-making, and citizen representation.

Each hypothesis is evaluated using:

$$P(h|t) = \text{softmax}(\text{model}(t, h))$$

The text passes the relevance filter if:

$$\max_i P(h_i|t) \geq 0.6$$

If the threshold is not met, the tool returns:

```
score = NA
interpretation = "The tool cannot infer a political stance from this text."
```

This prevents scoring unrelated content.

3.5 Hypothesis-Based Entailment Scoring

3.5.1 Natural Language Inference

Each input text is evaluated against a predefined set of theoretical hypotheses representing populist and pluralist democratic principles. A transformer-based Natural Language Inference (NLI) model estimates the probability that the input text semantically entails each hypothesis.

Formally, given a text t and hypothesis h, the model estimates:

$$P(h|t) = \text{softmax}(\text{model}(t, h))$$

where the entailment probability is interpreted as the degree to which the text supports the normative statement represented by the hypothesis.

This approach allows the tool to detect rhetorical meaning beyond surface-level keywords by evaluating semantic alignment between the text and theoretical claims about democracy.

3.5.2 Hypothesis Construction

Hypotheses are derived from political theory and empirical research on each dimension.

3.5.3 Illustrative Populist Hypotheses

Examples of populist hypotheses used for inference include:

```
"The author of this text believes corrupt politicians have betrayed ordinary
    working people": (1.0, "populist"),

"The author of this text believes citizens should vote directly on major issues
    instead of trusting representatives": (1.0, "populist"),

"The author of this text believes elites are out of touch with regular voters":
    (1.0, "populist"),

"The author of this text believes the political system is rigged to benefit
    wealthy donors": (1.0, "populist"),

"The author of this text believes ordinary citizens have more common sense than
    political experts": (1.0, "populist"),

"The author of this text believes career politicians care more about donors than
    ordinary voters": (1.0, "populist"),

"The author of this text believes the media and establishment work together
    against the people": (1.0, "populist"),

"The author of this text believes voters share the same basic values and
    priorities": (1.0, "populist"),
```

These hypotheses capture core dimensions of populist democratic rhetoric, including anti-elitism, majoritarianism, and the moral unity of the people.

3.5.4 Illustrative Pluralist Hypotheses

Examples of pluralist hypotheses include:

```
"The author of this text believes different communities have legitimate but
    conflicting needs": (1.0, "pluralist"),

"The author of this text believes political negotiations and compromises are a
    core democratic principle": (1.0, "pluralist"),

"The author of this text believes constitutional courts should protect minority
    rights from majority rule": (1.0, "pluralist"),

"The author of this text believes policy experts provide valuable technical
    knowledge to lawmakers": (1.0, "pluralist"),

"The author of this text believes democratic institutions have evolved to serve
    important functions": (1.0, "pluralist"),

"The author of this text believes elected representatives should balance
    constituent demands with broader considerations": (1.0, "pluralist"),
```

These hypotheses reflect principles of institutional constraint, diversity of interests, and procedural legitimacy.

3.5.5 Hypothesis Evaluation Procedure

For each input text:

1. The text is paired with each hypothesis.
2. The NLI model computes the entailment probability.
3. Probabilities are stored separately for each hypothesis set.

The output is a vector of semantic support values:

$$\mathbf{p}_{pop} = (p_1, \ldots, p_n)$$
$$\mathbf{p}_{plu} = (p_1, \ldots, p_m)$$

These values are later aggregated to produce the final score.

3.5.6 Implementation Example

```
def _get_entailment_prob(self, text, hypothesis):
    inputs = self.tokenizer(
        text,
        hypothesis,
        return_tensors="pt",
        truncation=True,
        max_length=512
    )
    with torch.no_grad():
        outputs = self.model(**inputs)
        prob = torch.softmax(outputs.logits, dim=-1)[0,
               self.entailment_idx].item()
    return prob
```

The output represents semantic support for the hypothesis.

3.6 Score Calculation

3.6.1 Average Hypothesis Support

Average support for each category is computed:

$$P_{pop} = \frac{1}{n} \sum_{i=1}^{n} p_i$$

$$P_{plu} = \frac{1}{m} \sum_{j=1}^{m} p_j$$

3.6.2 Difference-Based Score

The relative orientation of the text is:

$$D = P_{pop} - P_{plu}$$

The final score is scaled to a 0–10 range:

$$Score = 5 + 5D$$

The score is clipped:

$$Score \in [0, 10]$$

3.6.3 Implementation

```
difference = populist_avg - pluralist_avg
final_score = 5 + (difference * 5)
final_score = np.clip(final_score, 0, 10)
```

3.7 Score Interpretation

Populism vs Pluralism

Scores	Interpretation
0–2	Strong pluralist
2–4	Pluralist
4–6	Moderate
6–8	Populist
8–10	Strong populist

Economic Left vs Right

Scores	Interpretation
0–2	Far Left
2–4	Left
4–6	Center
6–8	Right
8–10	Far Right

Liberal Democracy

Scores	Interpretation
0–2	Strongly Opposes
2–4	Opposes
4–6	Moderate
6–8	Supports
8–10	Strongly Supports

3.8 Confidence Estimation

Confidence measures internal consistency of model predictions.

3.8.1 Variance-Based Reliability

Lower variance indicates greater consistency:

$$C = \frac{1}{1 + 4 \cdot Var}$$

Computed separately for each prediction.

3.8.2 Contradiction Detection

Contradiction is detected when both orientations receive strong support.

Procedure:

1. Select top-K hypothesis probabilities from each category.
2. Compute their averages.
3. Detect contradiction if:

$$\min(\overline{TopPop}, \overline{TopPlu}) > 0.25$$

3.8.3 Confidence Penalty

If contradiction exists:

$$C_{final} = C_{base} \times (1 - 0.8 \cdot penalty)$$

3.9 Output Structure

Each analysis produces:

- Score
- Confidence level
- Interpretation category
- Contradiction status
- Supported hypotheses

Example output:

```
{
  "score": 7.2,
  "confidence": 0.81,
  "interpretation": "Populist",
  "contradiction_detected": False
}
```

3.10 Validation

To assess the empirical validity of the hypothesis-based scoring approach, we compare model-generated scores with expert-coded evaluations of political texts.

The validation framework consists of three components:

3.10.1 Expert Evaluation Benchmark

Validation relies on human expert assessments as a reference standard. A sample of political texts was collected from multiple sources, including party manifestos, speeches, and social media posts. Texts were manually reviewed to ensure relevance for the theoretical constructs being measured.

A representative sample of 20 texts per dimension was selected. The evaluated dimensions included:

- Populism vs. pluralism
- Economic left vs. right ideology
- Support for liberal democratic values

Expert evaluators were recruited from authors who had published articles on these topics in leading political science journals (e.g., Political Communication, American Political Science Review). Participants were presented with a random selection of texts and asked to rate each text using a continuous scale representing the relevant political dimension.

In total, 147 expert responses were collected:

- 71 responses for populism/pluralism

- 37 responses for left/right ideology
- 38 responses for liberal democracy

The expert scores provide a benchmark against which model predictions are compared. Check the reference paper for additional information about the results obtained.

3.11 Limitations

- Scores depend on hypotheses design.
- Model predictions may reflect training data biases.
- Results reflect rhetorical framing rather than intent.