Problem Statement 1: Multi-AI Agent System for Zero shot Stance Detection

Overview

Understanding whether an author supports, opposes, or remains neutral on a topic is crucial for applications such as media monitoring and political analysis. This challenge involves building a stance detection system using a multi-agent approach where different AI agents work together to analyze and determine stance.

Requirements

1. Create a system of specialized AI agents that collaborate to determine stance:
    - One agent focuses on sentiment analysis.
    - Another identifies argumentative structures.
    - A third detects subtle linguistic signals of stance.
    - A coordinator agent synthesizes these insights.
2. Your system must accept:
    - **Text:** A statement, comment, or passage to analyze.
    - **Topic:** A keyphrase representing the subject of interest.
3. The output must be one classification: **"Favor," "Against," or "Neutral."**

Example

Text: "The renewable energy subsidies may cost more upfront, but they're essential for transitioning to a sustainable future and ultimately saving money long-term."

Topic: "Renewable energy subsidies"

Expected Output: "Favor"

Reasoning: Multiple agents could identify the acknowledgment of a drawback but the stronger emphasis on benefits and necessity, indicating overall favor.

Ref: https://arxiv.org/pdf/2310.10467, https://github.com/tsinghua-fib-lab/COLA

Problem Statement 2: Mini GRU Implementation for Stance Detection

Overview

Social media platforms, news organizations, and market researchers need automated tools to determine the stance expressed in text. The challenge here is to build a stance detection system using Gated Recurrent Units (GRU) that can efficiently process sequences of text to determine stance toward a specific topic.

Requirements

1. Implement a MinGRU-based neural network architecture that:
    - Efficiently captures sequential dependencies in text.
    - Incorporates topic information effectively into the model.
    - Handles variable-length inputs.
2. Your system must accept:
    - **Text:** A statement, comment, or passage to analyze.
    - **Topic:** A keyphrase representing the subject of interest.
3. The output must be one classification: **"Favor," "Against," or "Neutral."**

Example

Text: "Despite what some activists claim, the proposed data privacy regulation would stifle innovation and create unnecessary bureaucratic hurdles for startups."

Topic: "Data privacy regulation"

Expected Output: "Against"

Reasoning: The GRU should capture the negative framing toward the regulation despite the initial acknowledgment of opposing viewpoints.
References : https://arxiv.org/pdf/2410.01201

Problem Statement 3: Mini LSTM Implementation for Stance Detection

Overview

Stance detection is essential for content moderation, brand monitoring, and public opinion analysis. This challenge involves developing a stance detection system using Long Short-Term Memory (LSTM) networks to detect both explicit and implicit stance signals in text.

Requirements

1. Design an mini LSTM-based architecture that:
    - Effectively learns long-range dependencies in text.
    - Handles the relationship between text and topic.
    - Manages the challenge of implicit stance detection.
2. Your system must accept:
    - **Text:** A statement, comment, or passage to analyze.
    - **Topic:** A keyphrase representing the subject of interest.
3. The output must be one classification: **"Favor," "Against," or "Neutral."**

Example

Text: "The company announced its new AI features yesterday. The demo looked impressive, though there's always concern about how these technologies will impact jobs and privacy."

Topic: "Artificial intelligence"

Expected Output: "Neutral"

Reasoning: The min LSTM should identify the balanced presentation of both positive aspects ("impressive") and concerns, indicating a neutral stance rather than clear favor or against.

References : https://arxiv.org/pdf/2410.01201

Problem Statement 4: Chain-of-Thought Prompting for Zero Shot Stance Detection

Overview

Zero-shot stance detection often struggles with implicit reasoning and target generalization. This challenge involves developing a ZSSD model using **Chain-of-Thought (CoT) prompting**, enabling step-by-step reasoning to improve stance classification.

Requirements

1. Implement a **Chain-of-Thought (CoT) prompting** mechanism that:
    - Generates **intermediate reasoning steps** before predicting stance.
    - Enhances stance generalization for **unseen topics**.
    - Uses **few-shot or zero-shot prompting** to improve adaptability.
2. Your system must accept:
    - **Text:** A statement, comment, or passage to analyze.
    - **Topic:** A keyphrase representing the subject of interest.
3. The output must be one classification: **"Favor," "Against," or "Neutral."**

Example

Text: "Electric vehicles reduce carbon emissions significantly, making them the future of transportation."

Topic: "Fossil fuels"

Expected Output: "Against"

Reasoning: The CoT model identifies the positive stance toward electric vehicles, which implies a negative stance towards fossil fuels, leading to an "Against" classification.

Problem Statement 5: Instruction-Finetuned Zero Shot Stance Detection

Overview

Traditional fine-tuning struggles with zero-shot stance generalization. This challenge involves developing a stance detection model using **instruction fine-tuning**, where a model learns from diverse stance-related instructions rather than labeled examples.

Requirements

1. Train a **pretrained model (e.g., FLAN-T5, LLaMA 2)** with **stance detection-specific instructions**, including:
    - **Meta-task learning** from related NLP tasks (entailment, sentiment analysis, fact-checking).
    - **Natural language task descriptions** to improve target adaptability.
2. Your system must accept:
    - **Text:** A statement, comment, or passage to analyze.
    - **Topic:** A keyphrase representing the subject of interest.
3. The output must be one classification: **"Favor," "Against," or "Neutral."**

Example

Text: "The proposed tax cuts will help small businesses thrive."

Topic: "Corporate Taxation"

Expected Output: "Favor"

Reasoning: The instruction-tuned model understands the relation between tax cuts and small business growth, identifying a "Favor" stance.

Problem Statement 6: Alignment-Tuned Zero Shot Stance Detection

with Contrastive Learning

Overview

Zero-shot stance detection models often fail to align with **human reasoning** due to lack of explicit supervision. This challenge involves using **contrastive learning and Reinforcement Learning from Human Feedback (RLHF)** to improve stance consistency.

Requirements

1. Train a **contrastive stance model** that:
   - Uses **contrastive loss** to improve target-awareness in zero-shot settings.
   - Fine-tunes using **human-annotated stance feedback** via **RLHF**.
2. Your system must accept:
   - **Text:** A statement, comment, or passage to analyze.
   - **Topic:** A keyphrase representing the subject of interest.
3. The output must be one classification: **"Favor," "Against," or "Neutral."**

Example

Text: "Government subsidies for clean energy projects should be increased to combat climate change."

Topic: "Renewable Energy"

Expected Output: "Favor"

Reasoning: The contrastive model aligns with human reasoning by distinguishing support for clean energy from opposition to fossil fuels, classifying the stance as "Favor."

References “: https://arxiv.org/pdf/2212.10560

Problem Statement 7: Tree-of-Counterfactual Prompting for Zero Shot Stance Detection

Overview

Traditional stance detection lacks counterfactual reasoning, making it difficult to detect stance in complex scenarios. This challenge involves using **Tree-of-Counterfactual (ToC) Prompting**, where the model generates hypothetical counterarguments to improve zero-shot stance detection.

Requirements

1. Implement a **Tree-of-Counterfactual Prompting (ToC) model** that:
   - Generates **counterfactual versions** of the input text.
   - Uses **self-reflection prompting** to refine stance classification.
   - Applies **entailment-based stance verification** for improved accuracy.
2. Your system must accept:
   - **Text:** A statement, comment, or passage to analyze.
   - **Topic:** A keyphrase representing the subject of interest.
3. The output must be one classification: **"Favor," "Against," or "Neutral."**

Example

Text: "Cryptocurrencies provide financial freedom by reducing reliance on traditional banking institutions."

Topic: "Decentralized Finance"

Expected Output: "Favor"

Reasoning: The model generates a counterfactual version, such as "Cryptocurrencies lead to financial instability due to lack of regulation." By comparing the original and counterfactual claims, the system correctly classifies the stance as "Favor."

https://aclanthology.org/2024.acl-long.49.pdf

Problem Statement 8: Retrieval-Augmented Generation (RAG) for ZSSD

Overview

Zero-shot stance detection often fails due to **lack of explicit contextual knowledge**. This challenge involves implementing a **Retrieval-Augmented Generation (RAG) model**, which retrieves supporting information before stance classification.

Requirements

1. Develop a **RAG-based ZSSD model** that:
   - Retrieves **relevant documents** from Wikipedia, news archives, or domain-specific corpora.
   - Uses **cross-attention mechanisms** to incorporate external knowledge into stance prediction.
   - Generates **factually grounded explanations** before classifying stance.
2. The system must accept:
   - **Text:** A statement, comment, or passage to analyze.
   - **Topic:** A keyphrase representing the subject of interest.
3. The output must be one classification: **"Favor," "Against," or "Neutral."**

Example

Text: "Banning plastic bags is necessary to reduce environmental damage and protect marine life."

Topic: "Plastic Ban"

Expected Output: "Favor"

Reasoning: The RAG model retrieves **scientific studies and environmental policies** supporting the plastic ban, reinforcing the "Favor" classification.

Problem Statement 9: AI-Driven Precision Diagnostics for Resource-Limited Settings

Overview:
Many resource-limited healthcare settings lack affordable diagnostic tools, delaying the early detection of diseases such as diabetes. The challenge is to design an AI-powered risk prediction system that integrates secure and interoperable health data logging, ensuring privacy and explainability in rural environments.

Requirements:

1. Develop an AI-driven health diagnostics system that:
 - Predicts disease risk (e.g., diabetes) using clinical and environmental data.
 - Generates explainable outputs for healthcare workers.
 - Supports secure data logging to maintain patient privacy.
 - Works offline on low-cost devices (≤₹2,000).

Example Use Case:

A rural healthcare worker uses the AI model to assess diabetes risk in patients using a low-cost mobile device, receiving personalized health recommendations for early intervention.

Problem Statement 10: AI-Powered Vernacular Chatbots for Low-Cost Diagnostics

Overview:
Healthcare services in rural areas face infrastructure limitations and language barriers, making patient communication and symptom triage difficult. This challenge involves developing an AI-powered vernacular chatbot that assists in symptom analysis, patient education, and diagnostic recommendations for low-resource healthcare settings.

Requirements:

1. Develop an AI-powered chatbot that:
 - Operates in multiple regional languages.
 - Provides symptom-based triage and patient guidance.
 - Works on low-cost mobile devices with minimal infrastructure.
 - Ensures secure data management and patient privacy.

Example Use Case:

A farmer in a remote village reports symptoms to the chatbot in his native language. The system provides symptom analysis and recommends seeking medical attention, ensuring faster access to healthcare advice.

Problem Statement 11: AI-Optimized Genomic Data Pipelines for Indian Population Diversity

Overview:
India's vast genetic diversity is underrepresented in global genomic datasets, reducing the impact of precision medicine. This challenge focuses on developing a secure AI-driven genomic data pipeline that integrates clinical and genetic information, ensuring data privacy, interoperability, and real-world applicability in healthcare settings.

Requirements:

1. Develop an AI-powered genomic data pipeline that:
 - Integrates clinical and genomic datasets for disease risk prediction.
 - Ensures privacy and security in data sharing.
 - Supports seamless interoperability among healthcare institutions.
 - Addresses infrastructure challenges in low-resource settings.

Example Use Case:

A genomics research lab in India uses the AI-powered pipeline to analyze genetic markers for diabetes and heart disease, allowing better personalized healthcare treatments.

Problem Statement 12: Secure Blockchain-Enabled Health Data Exchange

Overview:
Health data exchange faces security risks, interoperability issues, and real-time accessibility challenges, especially in low-resource healthcare environments. This problem involves building a blockchain-powered health data exchange system that ensures secure, tamper-proof, and AI-assisted health record management.

Requirements:

1. Develop a blockchain-powered health data system that:
- Ensures tamper-proof patient data logging.
- Uses AI for real-time anomaly detection and data integrity monitoring.
- Supports encrypted health data sharing across institutions.
- Optimized for low-resource healthcare settings.

Example Use Case:

A government hospital network uses the system to securely share patient health data across multiple hospitals while detecting fraudulent activities, ensuring better patient care and privacy.