Contents

ADR-004: NLI Entailment Over Cosine Similarity

Status

Accepted

Context

Cosine similarity between embeddings is commonly used to measure semantic similarity. However, it has a critical flaw: **it cannot detect negation**.

The Negation Blindness Problem

```
Sentence A: "The Earth is flat"
Sentence B: "The Earth is not flat"

Cosine Similarity: 0.92 (highly similar!)
NLI Classification: CONTRADICTION
```

When embeddings are created, they capture semantic content without preserving logical relationships. The words "flat" and "not flat" refer to the same concept, so embeddings place them close together.

Impact on Cato

If Cato uses cosine similarity for surprise measurement: 1. Prediction: "X is true" 2. Outcome: "X is false" 3. Cosine similarity: HIGH (similar embeddings) 4. Surprise score: LOW (incorrectly!) 5.

No learning occurs despite being completely wrong

This is catastrophic for a learning system.

Decision

Use **Natural Language Inference (NLI)** with DeBERTa-large-MNLI for all surprise and consistency measurements.

NLI Classification

Label	Meaning	Surprise Score
ENTAILMENT	A implies B	0.0 (expected)
NEUTRAL	A neither implies nor contradicts B	0.5 (uncertain)
CONTRADICTION	A contradicts B	1.0 (surprising)

Model Selection

DeBERTa-large-MNLI chosen for: - State-of-the-art NLI accuracy (91.3% on MNLI) - Efficient inference (~50ms on GPU) - Well-calibrated confidence scores - Apache 2.0 license

Architecture

```
                  Prediction + Outcome
Prediction: "Claude is made by Anthropic"
Outcome: "Anthropic created Claude"
```

```
                  NLI Classifier (DeBERTa)
Input: [CLS] Prediction [SEP] Outcome [SEP]
Output: {entailment: 0.95, neutral: 0.04, contradiction: 0.01}
```

```
                  Surprise Calculator
Label: ENTAILMENT
Confidence: 0.95
Surprise Score: 0.0 × (1 - 0.95) = 0.0
```

Implementation

SageMaker Multi-Model Endpoint

Deploy DeBERTa on SageMaker MME for cost-efficient GPU sharing:

```python
class NLIService:
    """NLI classification using DeBERTa on SageMaker MME."""

    def __init__(
        self,
        endpoint_name: str = "cato-nli-mme",
        region: str = "us-east-1"
    ):
        self.runtime = boto3.client(
            "sagemaker-runtime",
            region_name=region
        )
        self.endpoint_name = endpoint_name

    async def classify(
        self,
        premise: str,
        hypothesis: str
    ) -> NLIResult:
        """
        Classify relationship between premise and hypothesis.

        Args:
            premise: The reference text (prediction)
            hypothesis: The text to compare (outcome)

        Returns:
            NLIResult with label, scores, and surprise value
        """
        payload = {
            "inputs": {
                "premise": premise,
                "hypothesis": hypothesis
            }
        }

        response = self.runtime.invoke_endpoint(
            EndpointName=self.endpoint_name,
            ContentType="application/json",
            Body=json.dumps(payload),
            TargetModel="deberta-large-mnli.tar.gz"
        )

        result = json.loads(response["Body"].read())

        # Extract scores
        scores = {
            "entailment": result["scores"][0],
```

```
            "neutral": result["scores"][1],
            "contradiction": result["scores"][2]
        }

        # Determine label
        label = max(scores, key=scores.get)
        confidence = scores[label]

        # Calculate surprise
        if label == "entailment":
            surprise = 0.0
        elif label == "neutral":
            surprise = 0.5
        else:  # contradiction
            surprise = 1.0

        # Weight by confidence
        weighted_surprise = surprise * confidence + 0.5 * (1 - confidence)

        return NLIResult(
            label=label,
            scores=scores,
            confidence=confidence,
            surprise=weighted_surprise
        )
```

TypeScript Client

```typescript
interface NLIResult {
  label: 'entailment' | 'neutral' | 'contradiction';
  scores: {
    entailment: number;
    neutral: number;
    contradiction: number;
  };
  confidence: number;
  surprise: number;
}

class NLIClient {
  private readonly sagemakerRuntime: SageMakerRuntimeClient;
  private readonly endpointName: string;

  async classify(
    premise: string,
    hypothesis: string
  ): Promise<NLIResult> {
    const command = new InvokeEndpointCommand({
```

```
      EndpointName: this.endpointName,
      ContentType: 'application/json',
      Body: JSON.stringify({
        inputs: { premise, hypothesis }
      }),
      TargetModel: 'deberta-large-mnli.tar.gz'
    });

    const response = await this.sagemakerRuntime.send(command);
    const result = JSON.parse(
      new TextDecoder().decode(response.Body)
    );

    return this.parseResult(result);
  }
}
```

Consequences

Positive

- **Correct negation handling**: Contradictions detected accurately
- **Calibrated uncertainty**: NEUTRAL captures genuine uncertainty
- **Interpretable**: Three-way classification is human-understandable
- **Robust**: DeBERTa handles paraphrasing, synonyms, and complex logic

Negative

- **Additional latency**: ~50ms per classification
- **GPU requirement**: DeBERTa needs GPU for efficient inference
- **Hosting cost**: ~$200-500/month for SageMaker MME
- **Pair-wise limitation**: Can only compare two texts at a time

Comparison: Cosine vs NLI

Scenario	Cosine Similarity	NLI	Correct?
"X is true" vs "X is true"	1.0 (similar)	ENTAILMENT	Both
"X is true" vs "X is not true"	0.92 (similar)	CONTRADICTION	NLI
"Dogs are mammals" vs "Canines are warm-blooded"	0.65 (medium)	ENTAILMENT	NLI
"It's raining" vs "The weather is wet"	0.55 (medium)	ENTAILMENT	NLI

Scenario	Cosine Similarity	NLI	Correct?
"2+2=4" vs "The sum is four"	0.45 (low)	ENTAILMENT	NLI

Infrastructure

SageMaker MME Configuration

```
resource "aws_sagemaker_endpoint" "nli_mme" {
  name                 = "cato-nli-mme"
  endpoint_config_name = aws_sagemaker_endpoint_configuration.nli_mme.name
}

resource "aws_sagemaker_endpoint_configuration" "nli_mme" {
  name = "cato-nli-mme-config"

  production_variants {
    variant_name           = "primary"
    model_name             = aws_sagemaker_model.nli_mme.name
    instance_type          = "ml.g4dn.xlarge"  # Cost-effective GPU
    initial_instance_count = 2

    # Multi-model endpoint settings
    container_startup_health_check_timeout_in_seconds = 600
  }
}
```

Scaling

Users	NLI Calls/sec	Instances	Cost/month
100K	10	2	$200
1M	100	5	$500
10M	500	20	$2,000

References

- DeBERTa: Decoding-enhanced BERT with Disentangled Attention
- MNLI: Multi-Genre Natural Language Inference
- SageMaker Multi-Model Endpoints