Semantic Classification of Math Problems Using NLP

Group Final Project Report

Course: NLP Final Project (Amir Jafari)

Authors: Phanindra Kalaga; Prudhvi Chekuri; Deepika Reddygari

---

1. Motivation & Project Overview

Manually tagging math problems into domains delays content curation for adaptive learning. We aimed to automate this by building a pipeline that “understands” both natural language and embedded LaTeX notation, assigning each problem to one of eight categories: Algebra, Arithmetic, Calculus, Geometry, Number Theory, Probability, Statistics, and Trigonometry . Our approach had two phases:

1. **Classical baselines**: TF-IDF features (word & char n-grams) with Multinomial Naive Bayes, Random Forest, and XGBoost.

2. **Advanced methods**: Transformer fine-tuning (MathBERT, DeBERTa, T5, Llama) and a hard-voting ensemble.

---

2. Dataset Description

- **Source**: KAUST Math Problem Classification (25 000 train / 5 000 test)

- **Classes**: 8 balanced topics (~3 125 examples each)

- **Characteristics & Challenges**:

    - LaTeX spans (`$…$`) in ~40% of examples

    - Varied question lengths (short prompts to multi-sentence problems)

    - Occasional noise (scraping artifacts)

---

3. Group Methodology Overview

1. **Preprocessing & Feature Engineering**
   - Normalized all `$…$` spans to the token **MATH**, lowercased, removed non-alphanumeric characters, and collapsed whitespace.
   - Extracted **word n-grams** (1–2) and **char n-grams** (3–5) with TF-IDF weighting.
2. **Classical Baselines**
   - **Naive Bayes** (`α=1.0`)
   - **Random Forest** (100 trees, `class_weight='balanced'`)
   - **XGBoost** (`max_depth=6`, `learning_rate=0.1`, `subsample=0.8`, `colsample_bytree=0.8`)
3. **Transformer Models**
   - **MathBERT**: domain-pretrained BERT variant.
   - **DeBERTa-V3 base**: disentangled attention for richer contextualization.
   - **T5-base**: seq2seq framing for classification.
   - **Llama-3.2 1B**: instruction-tuned via LoRA with 4-bit quantization.
4. **Ensembling**
   - Hard voting among DeBERTa, T5, and Llama, with DeBERTa as tie-breaker.
5. **Deployment & Demo**
   - Streamlit app and Docker container orchestrated by Phanindra for live classification demos.

---

4. Individual Contributions

4.1 Phanindra Kalaga

- Orchestrated project setup: scripts for data/model retrieval (`get_assets.sh`), demo launch (`run_demo.sh`), and Streamlit app (`app.py`).
- Packaged dependencies (`requirements.txt`), containerized the demo for AWS deployment .

4.2 Prudhvi Chekuri

- Fine-tuned transformer architectures: DeBERTa-V3 for sequence classification, T5 for text-to-text classification, and on-device Llama-3.2 1B via LoRA.
- Developed ensemble notebook to combine transformer outputs via hard voting .

4.3 Deepika Reddygari

- Built classical baseline pipelines: `preprocessing.py`, `features.py`, `train_baselines.py`, `evaluate.py`.
- Executed grid searches for TF-IDF and XGBoost hyperparameters; conducted error analysis with confusion matrices; ensured reproducibility with fixed seeds and experiment logs .

---

5. Consolidated Results:

Model	Train Time	F1-Micro (Train)	F1-Micro (Private Test)
Multinomial Naive Bayes	1.04 m	0.72	0.704
Random Forest Classifier	6.51 m	0.74	0.7388

XGBoost	10.02 m	0.79	0.7678
LightGBM	6.24 m	0.789	0.7862
MathBERT	50 m	0.83	0.8152
Llama-1b	1 h 30 m	0.85	0.8346
T5	2 h	0.83	0.8239
DeBERTa	4 h	0.9	0.851
Ensemble (Llama + T5 + DeBERTa)– Hard Voting –	–	0.8588	0.8588

Table1: Performance across all models.

---

6. Discussion & Key Insights

- **Classical vs. Advanced:** Classical baselines (up to 0.7678) train in seconds; transformer models (up to 0.8510) require hours but yield >8 pt F1 gains.
- **Feature Impact:** Character-level TF-IDF (XGBoost) outperforms word-only TF-IDF by ~3 F1 points.
- **Ensembling Benefit:** Hard voting adds ~0.008 F1 over the best single transformer.
- **Compute Trade-off:** Efficient mixed-precision and PEFT (LoRA) make large-model training feasible on limited hardware.

---

7. Future Work

- **Model Distillation:** Compress the ensemble into a lightweight student model for real-time inference.
- **Advanced Hyperparameter Search:** Employ Bayesian optimization or random search for both classical and transformer pipelines .

- **Data Augmentation:** Incorporate back-translation and equation paraphrasing to improve robustness to notation variations.
- **Robust Evaluation:** Implement k-fold cross-validation across all models to reduce selection bias.

---

8. References

1. Final Project Guidelines, NLP Final Project
2. Project Proposal: Semantic Classification of Math Problems Using NLP
3. Phanindra Kalaga, Individual Final Project Report
4. Prudhvi Chekuri, Individual Final Project Report
5. Deepika Reddygari, Individual Final Project Report
6. Presentation: Classification of Mathematical Problems – NLP Approaches
7. Sanh *et al.* (2019). DistilBERT: Smaller, Faster, Cheaper and Lighter. arXiv:1910.01108.
8. He *et al.* (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. arXiv:2006.03654.
9. Raffel *et al.* (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. JMLR, 21(140).

End of Group Final Report