1 Dataset

Inspired by paper "Pretraining on the Test Set Is All You Need" Schaeffer [2023] I have decided to create dataset without using any already existing datasets like GYAFC Rao and Tetreault [2018] or XFORMAL Briakou et al. [2021], because most of models made to detect sentence formality are fine-tuned on those datasets, thus making experiments untrustworthy. Dataset contains 1200 sentences split into 4 categories,

Category	Type	Example
Education	Formal	A catalyst lowers activation energy in reactions.
	Informal	A catalyst helps reactions happen easier.
Emergency	Formal	A tornado has been spotted in your county; take shelter in your basement immediately.
	Informal	Wildfire's getting closer—if you're in the west side, grab your stuff and go.
Guides	Formal	To improve reliability, distribute your API calls across multiple availability zones when possible.
	Informal	When sending the same request multiple times, use an idempotency key so we don't duplicate stuff.
Workplace	Formal	Please provide your availability for a 30-minute call to discuss the proposal.
	Informal	When are you free for a quick chat about the proposal? 30 mins max.

Table 1: Formal and Informal Text Examples by Category

Every category has been split into 3 subcategories; small with 5-10 words, medium with 11-20 words, and long with 21+ words, to compare methods accuracy on different context sizes.

The dataset has been fully artificially generated by deepseek-v3. For example here is prompt for emergency sentences:

```
Generate 25 formal and 25 informal examples for emergency communications.  Ensure the
formal examples sound official (e.g., government alerts), while the informal ones are
casual warnings (e.g., texting a friend).
Avoid emojis.  All of them should be 5-10 words.
Write all of them in this way:  sentence;formal/informal
Allow some variety in formal ones, they shouldn't be too similar, and informal ones
should be visibly informal, but you may generate some harder to spot ones
```

2 Methodology

Transformer-Based Encoders

This group included several models fine-tuned for formality detection, taken from Dementieva et al. [2023]:

- **Deberta large ranker**: A 406M parameter model based on DeBERTa (large), fine-tuned on the GYAFC dataset.
- **mdistilbert ranker**: A 135M parameter model based on mDistilBERT, fine-tuned on the X-FORMAL dataset.
- **xlmr formality classifier**: A 278M parameter classifier based on XLM-Roberta, trained on the XFORMAL dataset.

LLM-Based Methods

A selection of LLMs which I was able to get to run(more in problems section):

- **llama3.2-11b-vision** (11B parameters)
- **llama3.3-70b** (70B parameters)
- **deepseek-v3** (671B parameters)
- **chat gpt 4o-mini** (Unknown parameter count)
- **chat gpt 4o** (Estimated ˜200B parameters)

I tested LLMs on zero-shot scenario with temperature=0, The prompt provided to every LLM was the same and is shown below:

```
Analyze the formality of the following sentence on a scale of 0 to 100, where 0 is extremely
informal (e.g., slang, casual chat) and 100 is highly formal (e.g., academic/professional
writing). Consider:

  • Vocabulary (slang vs. technical terms)
  • Grammar (contractions, sentence structure)
  • Tone (casual vs. respectful/professional)

Return only the numerical score (0-100) with no explanation.
Sentence: {sentence}
```

Baseline Model

As a baseline I have decided to use pretrained bert-base-uncased, and I have chosen random 10 formal and their 10 informal counterparts and calculated average formal embedding and average informal embedding, using BERT last layer. Later I calculate cosine similarity between each token of sentence and both averages to predict how formal it is. Then i predict using average score of all tokens from sentence. For details look into baseline_model.py in provided implementation

3 Results

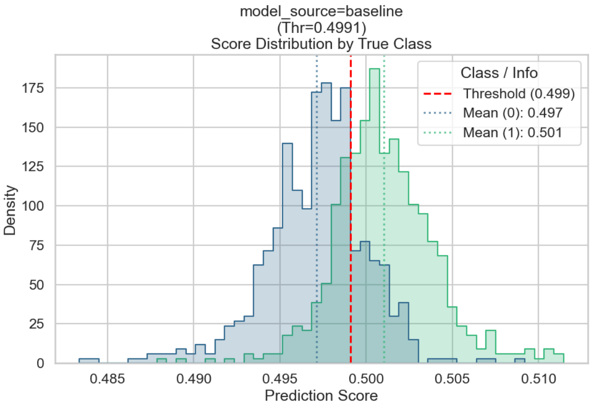


Figure 1: Baseline prediction distribution

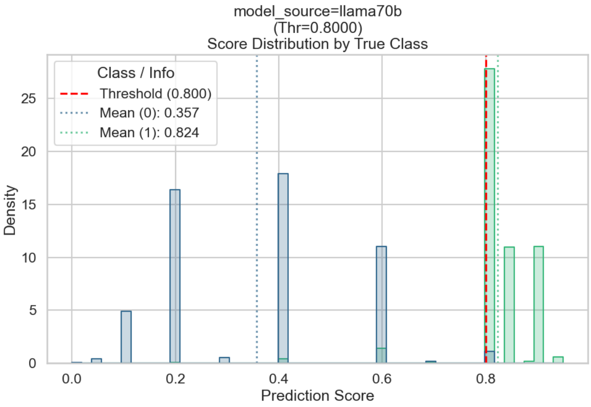


Figure 2: LLama 70B prediction distribution

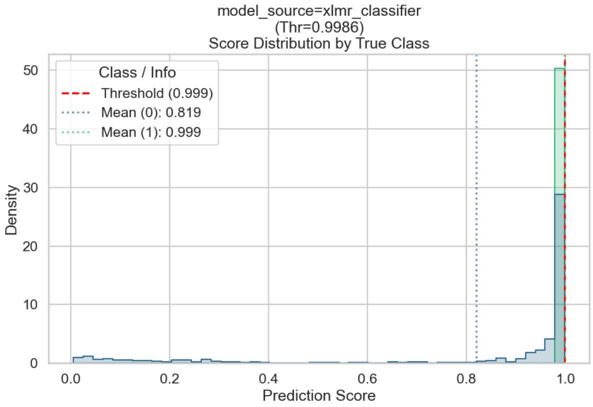
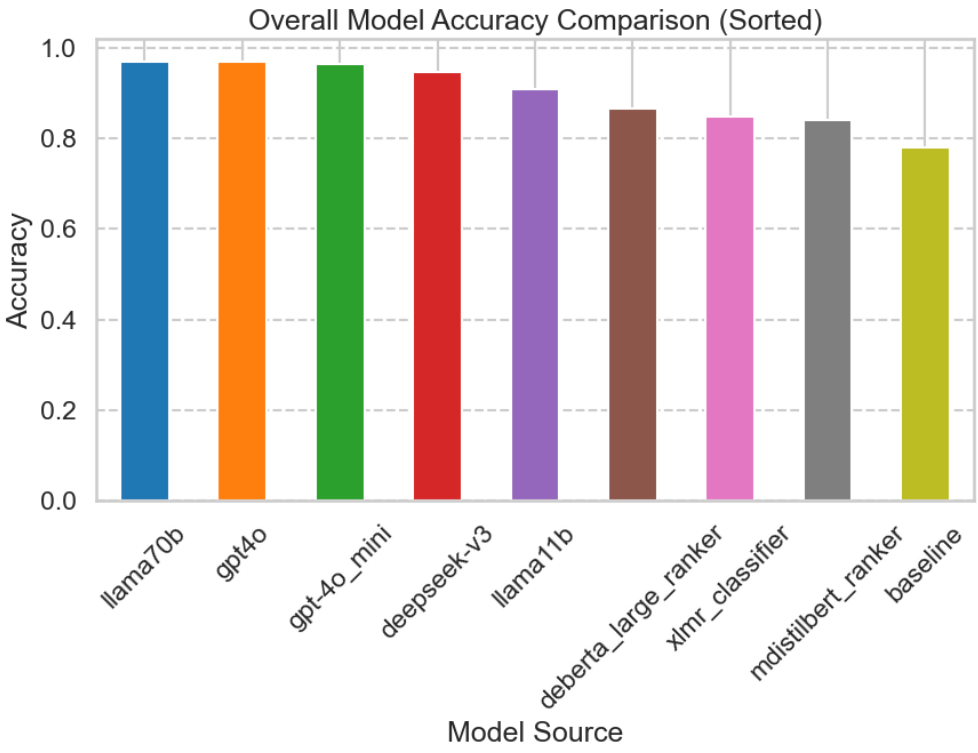


Figure 3: XLMR Classifier predictions

As models had various distributions of values instead of using fixed threshold I have decided to use point on ROC Curve that maximises Maximize Youden's J Statistic, as this method gave me the best results.



We can clearly see the correlation between number of parameters of model and accuracy. We also can see that the llama3.3-70b outperformed bigger models, but they all got very strong scores, with only deepsek-v3 getting visibly lower score. Our baseline was outperformed by all methods, but still got reasonably good score, as for such simple solution.

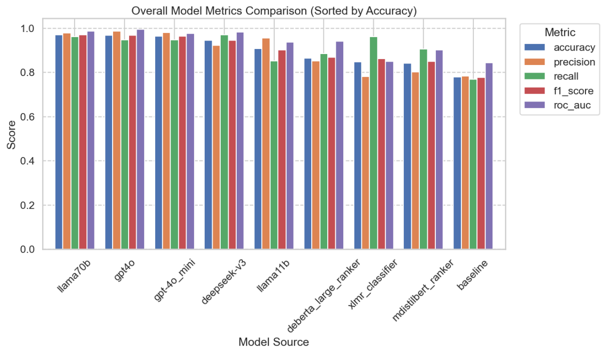


Figure 4: Model Metrics Comparison

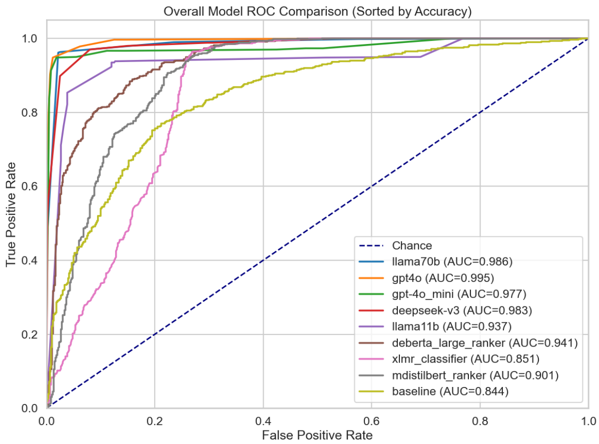


Figure 5: Model ROC Comparison

Figure 4 shows the Large Language Models (llama70b, gpt4o, gpt-4o_mini) performed best, hitting over 95% Accuracy. They achieved a good balance between Precision and Recall, leading to high F1 scores. Most models had good AUC scores too. In contrast, some fine-tuned models like xlmr_classifier focused heavily on finding almost all formal text (high Recall), even if it meant mislabeling more informal text (lower Precision). In Figure 5 we can see that the ROC curve looks very similarly on all LLM-based models, and very similarly on all Transformer-Encoder based models.

Table 2: Model Performance Metrics

Model	Threshold	Accuracy	Precision	Recall	F1-Score	AUC
llama70b	0.80000	0.97000	0.97797	0.96167	0.96975	0.98616
gpt4o	0.70000	0.96833	0.98785	0.94833	0.96769	0.99477
gpt-4o_mini	0.70000	0.96500	0.98103	0.94833	0.96441	0.97669
deepseek-v3	0.70000	0.94500	0.92381	0.97000	0.94634	0.98312
llama11b	0.74000	0.90750	0.95701	0.85333	0.90220	0.93717
deberta_large_ranker	0.99581	0.86583	0.85233	0.88500	0.86836	0.94123
xlmr_classifier	0.99860	0.84750	0.78214	0.96333	0.86333	0.85100
mdistilbert_ranker	0.99901	0.84167	0.80236	0.90667	0.85133	0.90126
baseline	0.49909	0.77917	0.78438	0.77000	0.77712	0.84376

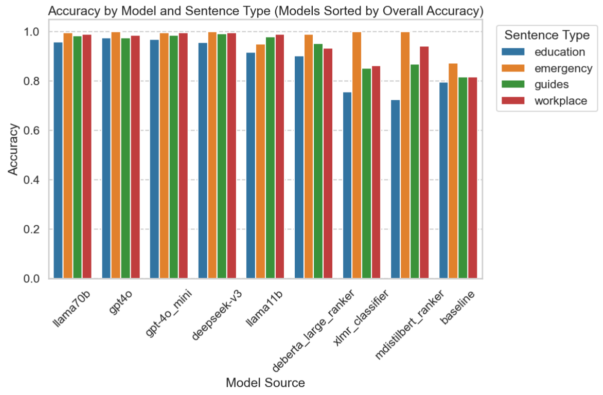


Figure 6: Scores distribution based on sentence type.

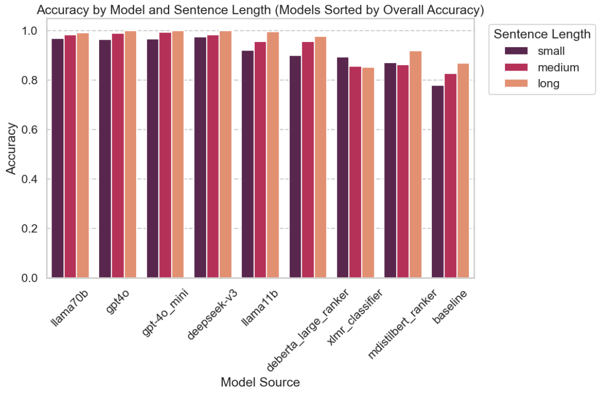


Figure 7: Scores distribution based on sentence length.

In Figure 6 we can see that LLM based solutions have similar accuracy across all categories while solutions fine-tuned/trained on GYAFC/XFORMAL have some visible lacks of knowledge, in some categories like education.

In Figure 7 we can see that almost all models benefit from longer sentences, the difference can be seen the best in mid-sized models, in my opinion this comes down to fact that they are too small to be able to catch formality immediately, and are big enough to have precious knowledge that helps them with working with more words.

4 Conclusions

I have created a evaluation system that tests how well formality detection model performs in various real-world scenarios, showing that GYAFC and XFORMAL datasets have some clear gaps, and shows how well tested model uses more data to get better accuracy.

5 Problems

While writing this project I had 2 main problems, that both did cost me combined a few hours.
1. llama api - As I was trying to get results from models other than chatgpt I stumbled upon a website called https://www.llmapi.com/ , which offered that they have API where you can after paying use loads of different LLM models, the only problem was, running any small model would cause server to return a specific error, and as I read the docs, and tried to debug the error and contacted owner on discord, but still have got no answer, so probably project is half-dead, as I have already spend too much time on this project, i decided that available LLMs will be enough
2. My perfectionism - As I wanted my method to actually have any viable application, and to be as good as possible, I have spent way too much time thinking reading about similar topics, and reiterating, for example,

into what categories should I split my dataset into, to get some good results, what categories could show lack of knowledge of model trained on dataset based on Yahoo Answers. Like maybe most definitions and science based stuff will be written in very formal way, and model will struggle in detecting informal ones, maybe guides, which are mostly written in informal way, etc. etc. but even though I spent about 2x recommended time, I am pretty happy with the results

References

E. Briakou, D. Lu, K. Zhang, and J. Tetreault. Xformal: A benchmark for multilingual formality style transfer, 2021. URL https://arxiv.org/abs/2104.04108.

D. Dementieva, N. Babakov, and A. Panchenko. Detecting text formality: A study of text classification approaches. In R. Mitkov and G. Angelova, editors, *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 274–284, Varna, Bulgaria, Sept. 2023. INCOMA Ltd., Shoumen, Bulgaria. URL https://aclanthology.org/2023.ranlp-1.31.

S. Rao and J. Tetreault. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer, 2018. URL https://arxiv.org/abs/1803.06535.

R. Schaeffer. Pretraining on the test set is all you need, 2023. URL https://arxiv.org/abs/2309.08632.