ME 5374-ST Fall 2025

Machine Learning for Materials Science and Discovery

Asst. Prof. Peter Schindler

Dr. Emad Rezaei

Large Language Models for Text Classification

Northeastern University

Text Classification

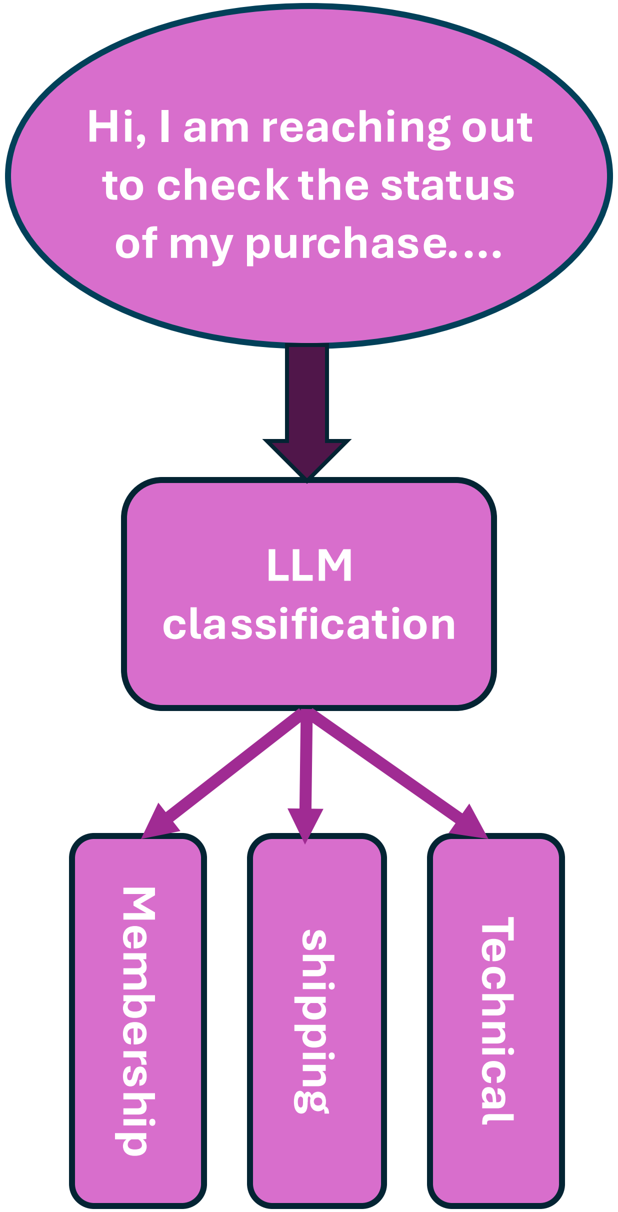
The goal of classification is to train a model to assign a label or class to some input text

Sentiment analysis ,intent detection, entity extraction, language detection and more

Large Language Models advantages over traditional ML models for text classification:

- ✓ Language understanding
- ✓ Reduce feature engineering
- ✓ Zero shot and few shot learning
- ✓ Handling diverse formats

Both representation and generative models are leveraged for text classification.



Classification with Representation Models

Two approaches: Task-specific model or Embedding model

Developed by fine-tuning a base model on a specific task.

Task-specific model: a representation model trained for a specific task.

Embedding model: general-purpose embeddings used for a variety of tasks.

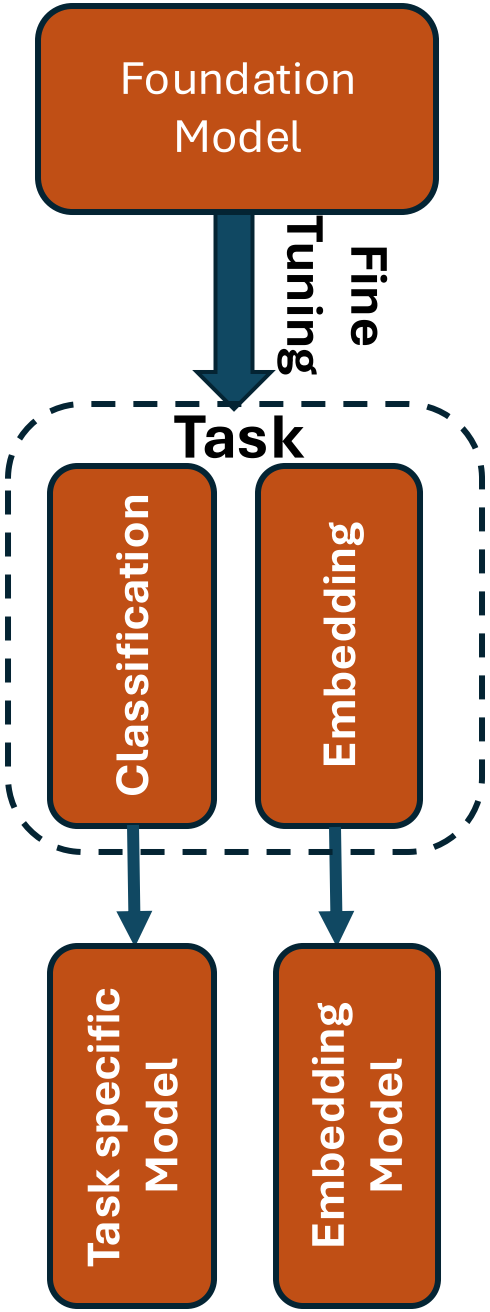
Over 60,000 models on the Hugging Face Hub for text classification!

How to select the best model???

Nearly impossible to test 60000 models!

Well-known models are great starting points e.g. BERT family

MTEB Leaderboard lists models benchmarked across various tasks



Task Specific Models

Developed with a well-defined focus and trained on domain-specific data.

✓ Higher accuracy

✓ Lower computational cost

✓ Lower risk of fabrication

X Requires task-specific training data

X Potential for outdated predictions

X High upfront cost

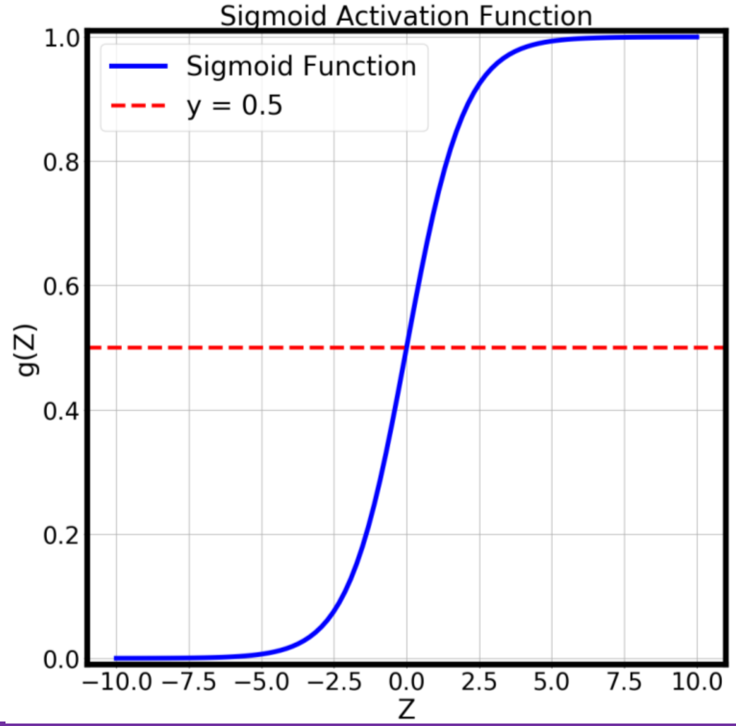
Embeddings for classification

Supervised classification

Instead of directly applying the representation model for classification, we can use an embedding model for developing features which will then be fed into a classifier (like logistic regression).

No need to fine-tune the embedding model and doable on CPUs.

1- Convert text input to embeddings by embedding models.
2- Embeddings serve as the input features to the classifier.

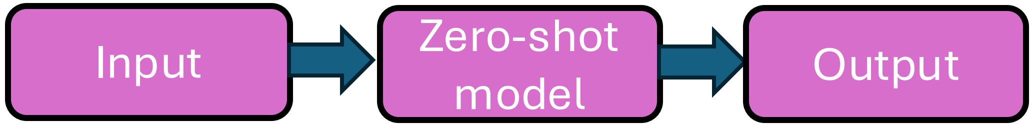


Embeddings for classification

Unsupervised classification

Zero-shot classification on unlabeled data to explore if the task seems doable.

Zero-shot classification predicts the labels of input text, although it was not trained on them.



Input → Zero-shot model → Output

How to do zero-shot classification with zero embeddings?

Our labels are described based on what they should represent.
Describing and embedding the labels and documents → data needed to work with

Text Classification with Generative Models

Sequence to sequence model:

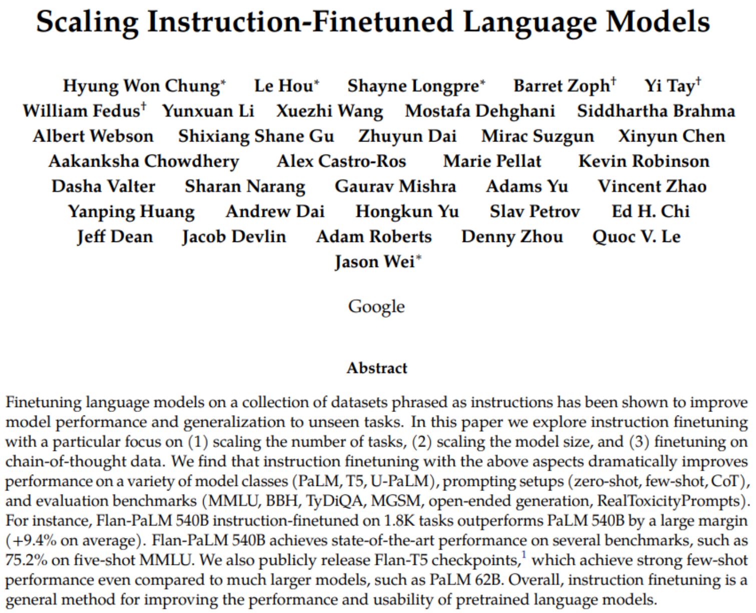
- Input = a sequence of tokens
- Model= Generative model to generate a text
- Output = a sequence of tokens

prompt engineering : help it understand the context and guide it toward the desired answer.

Examples of prompt (input) for movie reviews:

- What sentiment does this review have?
- Is this review negative or positive?
- Rate this sentiment.

Text-to-Text Transfer Transformer for Text classification



Scaling Instruction-Finetuned Language Models

Hyung Won Chung* Le Hou* Shayne Longpre* Barret Zoph† Yi Tay†
William Fedus† Yunxuan Li Xuezhi Wang Mostafa Dehghani Siddhartha Brahma
Albert Webson Shixiang Shane Gu Zhuyun Dai Mirac Suzgun Xinyun Chen
Aakanksha Chowdhery Alex Castro-Ros Marie Pellat Kevin Robinson
Dasha Valter Sharan Narang Gaurav Mishra Adams Yu Vincent Zhao
Yanping Huang Andrew Dai Hongkun Yu Slav Petrov Ed H. Chi
Jeff Dean Jacob Devlin Adam Roberts Denny Zhou Quoc V. Le
Jason Wei*

Google

Abstract

Finetuning language models on a collection of datasets phrased as instructions has been shown to improve model performance and generalization to unseen tasks. In this paper we explore instruction finetuning with a particular focus on (1) scaling the number of tasks, (2) scaling the model size, and (3) finetuning on chain-of-thought data. We find that instruction finetuning with the above aspects dramatically improves performance on a variety of model classes (PaLM, T5, U-PaLM), prompting setups (zero-shot, few-shot, CoT), and evaluation benchmarks (MMLU, BBH, TyDiQA, MGSM, open-ended generation, RealToxicityPrompts). For instance, Flan-PaLM 540B instruction-finetuned on 1.8K tasks outperforms PaLM 540B by a large margin (+9.4% on average). Flan-PaLM 540B achieves state-of-the-art performance on several benchmarks, such as 75.2% on five-shot MMLU. We also publicly release Flan-T5 checkpoints,[1] which achieve strong few-shot performance even compared to much larger models, such as PaLM 62B. Overall, instruction finetuning is a general method for improving the performance and usability of pretrained language models.

[2210.11416] Scaling Instruction-Finetuned Language Models

An encoder-decoder architecture.

Text-to-Text Transfer Transformer (T5).

Pre-trained using masked language modeling.

Instead of performing task-specific fine-tuning, the approach converts every task into a sequence-to-sequence format, allowing the model to be trained on all tasks simultaneously.

Exercise

- Lt's apply LLMs for text classification and evaluate the results.

- Google Colab

Trainable Models

Previously, we kept both model and classification head frozen. What if we let them be updated during training?

1. Start with a pre-trained BERT model and its tokenizer
2. A classification head to the output layer.
3. preprocessing text by tokenizing it, padding, and creating an attention mask.
4. Train the model by updating its parameters to fine-tune the classification head on your specific dataset
5. Evaluate the fine-tuned model

Few-Shot Classification

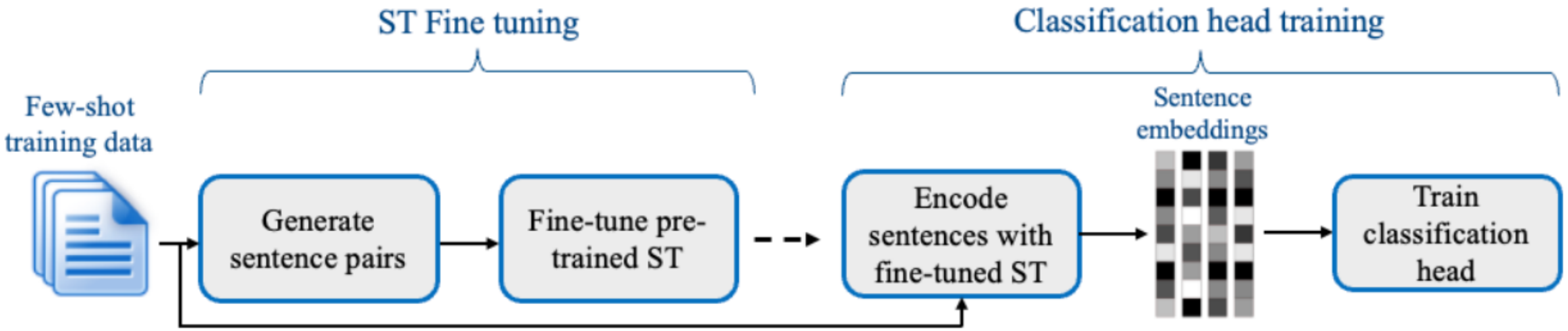
A supervised classification approach where the classifier learn target labels based on only a few labeled examples.

Useful when you have not enough labeled data points available for a classification task.

This approach takes a few high-quality labeled points which the model will be trained on.

The Algorithm based on 3 steps:

1. Sampling training data
2. Fine-tuning embeddings
3. Training a classifier



🤗 Models | 📊 Datasets | 📕 Documentation | 📖 Blog | 📃 Paper

SetFit - Efficient Few-shot Learning with Sentence Transformers

https://github.com/huggingface/setfit

Few-Shot Classification

Sentence	Class
I like soccer.	Sports
I used to play hockey.	Sports
Fe is ferromagnetic.	Materials Science
Zr has three phases.	Materials Science

Necessary data: in-class and out-class

Generate sentence pairs to fine-tune the embedding model → contrastive learning

Sentence 1	Sentence 2	Pair type
I like soccer.	Fe is ferromagnetic.	Negative
Zr has three phases.	I used to play hockey.	Negative
I like soccer.	I used to play hockey.	Positive
Zr has three phases.	Fe is ferromagnetic.	Positive

Utilize generated these pairs to fine-tune a Sentence Transformers model. It will create embeddings that are tuned to the classification task.

We create embeddings for all sentences which will serve as the input of a classifier.

The classifier learns from our finetuned embeddings to accurately predict unseen sentences.

References

1. Hands-On Large Language Models: Language Understanding and Generation, Book by Jay Alammar and Maarten Grootendorst.
2. Flexible, Model-Agnostic Method for Materials Data Extraction from Text Using General Purpose Language Models. Maciej P. Polak et.al.
3. Accelerating materials language processing with large language models. Choi et.al.
4. “Efficient few-shot learning without prompts.” Lewis Tunstall et al. arXiv preprint arXiv:2209.11055 (2022).