Procesamiento de Lenguaje Natural Avanzado

Fine-Tuning vs Feature Extraction

Biases and Limitations of Contextual Representations

Dra. Helena Gómez Adorno
helena.gomez@iimas.unam.mx

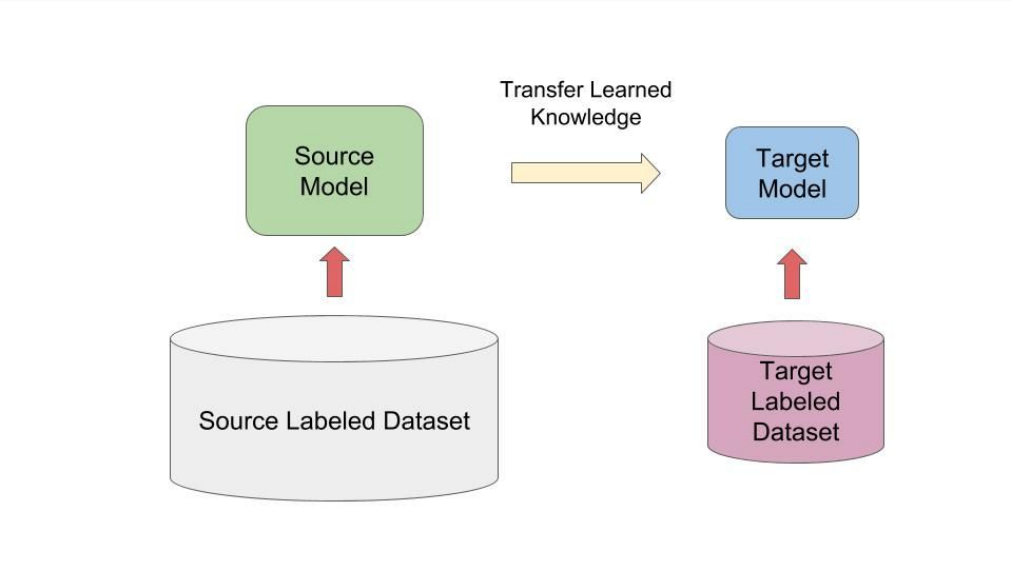
Dr. Fazlourrahman Balouchzahi
fbalouc@iimas.unam.mx

Correo del curso:
pln.cienciadedatos@gmail.com

- Pretrained transformer-based models such as BERT provide powerful contextual representations of language. These representations are learned through large-scale self-supervised objectives and capture syntactic, semantic, and contextual information across diverse corpora.
- **However, an important question arises in downstream applications:**
- Should we treat these pretrained representations as fixed feature extractors, or should we adapt the entire model to our specific task through fine-tuning?
- In this lecture, we will:
  - Compare feature extraction and full fine-tuning
  - Analyze their computational and representational implications
  - Examine risks such as overfitting and catastrophic forgetting
  - Discuss bias and limitations in contextual representations

---

Transformers



But what is transfer learning?

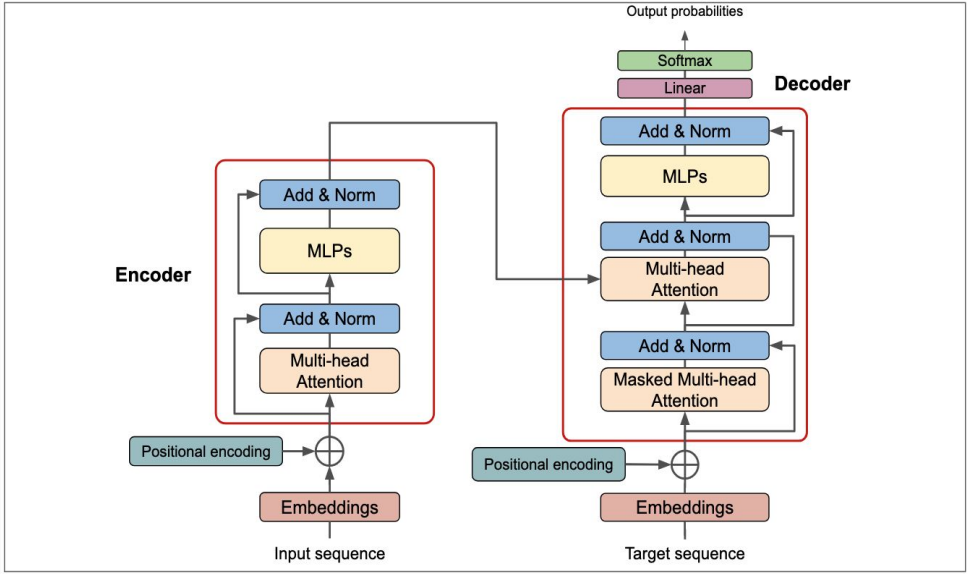
Modern transformer models are pretrained using self-supervised objectives such as:

- Masked Language Modeling (MLM)
- Next Sentence Prediction (NSP)
- Causal Language Modeling

During pretraining, the model learns general linguistic patterns from large corpora such as Wikipedia, BooksCorpus, or web-scale text.

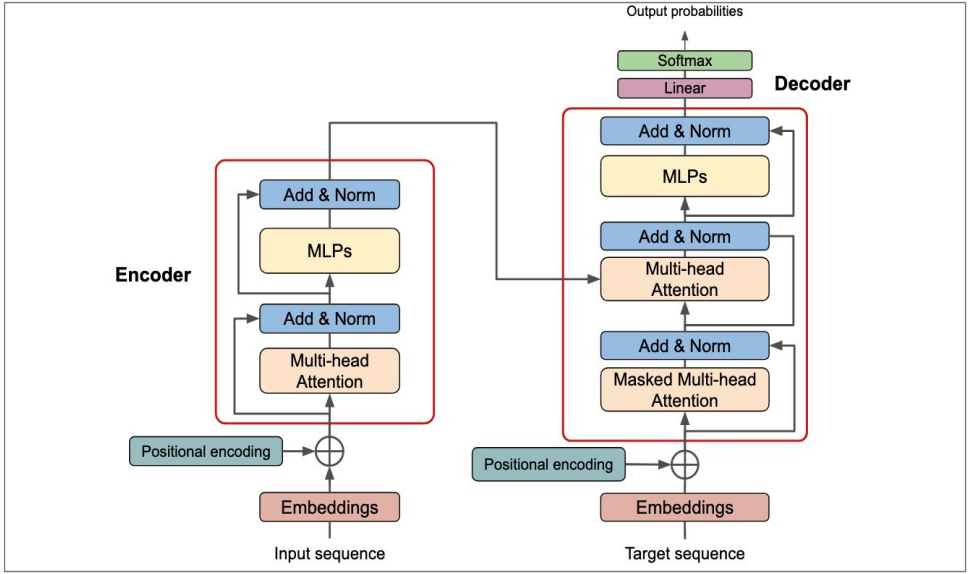
However, in a downstream supervised task (e.g., sentiment classification),

This creates a transfer learning scenario in which we must decide how much of the pretrained knowledge to preserve and how much to adapt.



From Pretraining to Downstream Tasks

Who will present us the layers in next session?



Task

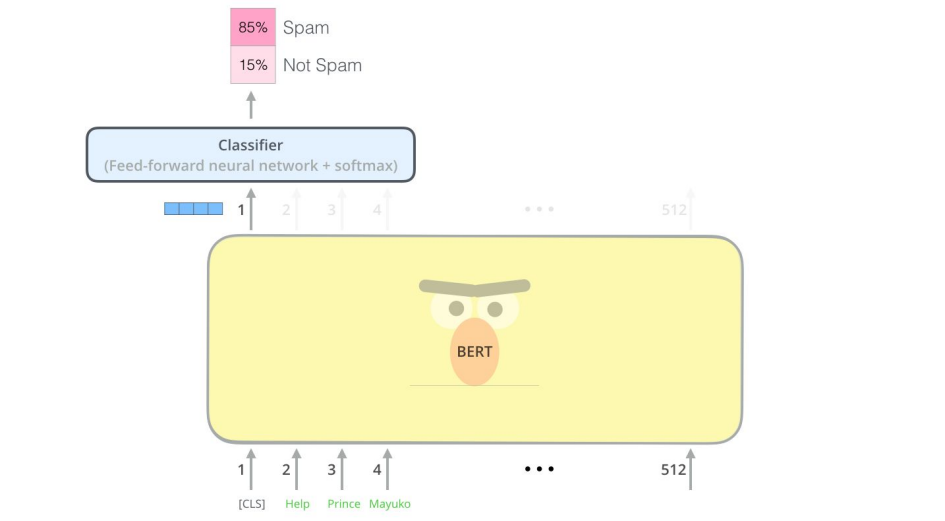
Transformer-based models produce contextual representations at multiple layers.

Each layer captures different types of linguistic information:

- **Lower layers:** syntactic structure, part-of-speech patterns
- **Middle layers:** phrase-level and compositional semantics
- **Higher layers:** more task-oriented abstractions

When using BERT for classification, the [CLS] token representation from the final layer is often used as a summary embedding.

- If we freeze the model, we preserve this learned hierarchy.
- If we fine-tune, we modify it.



What Does a Contextual Model Actually Learn?

Feature extraction is a transfer learning strategy in which the pretrained transformer model is kept fixed (all parameters are frozen), and only a lightweight classifier layer is trained on top of the extracted representations.

Formally, let:

$h = f_\theta(x)$

where f_θ is the pretrained transformer with parameters θ.

In feature extraction:

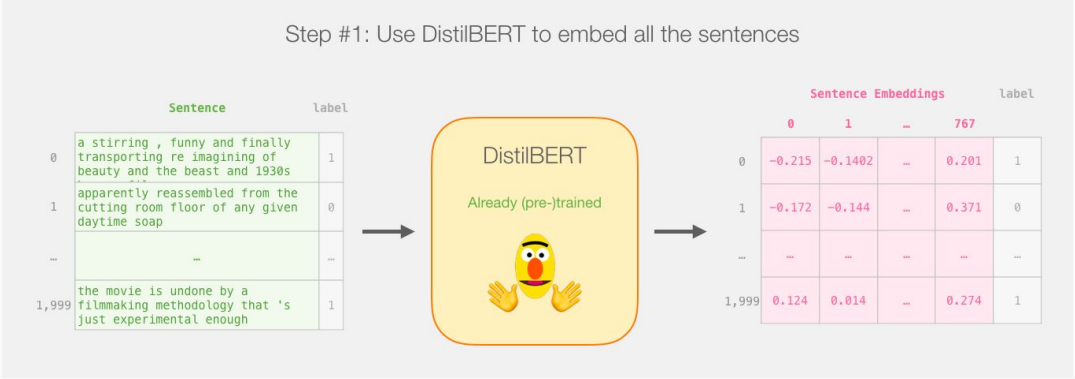
θ is fixed

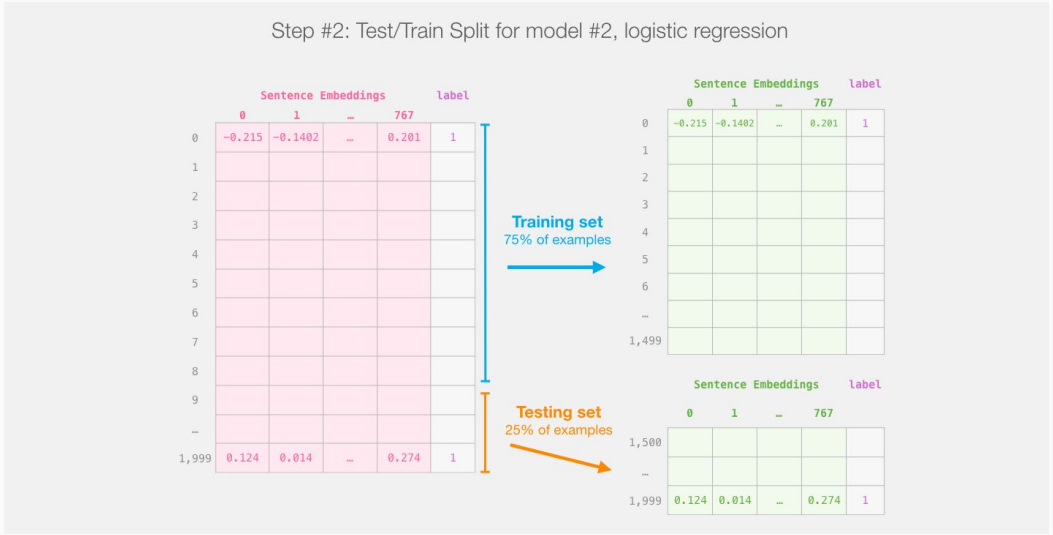
And we only train a new classifier:

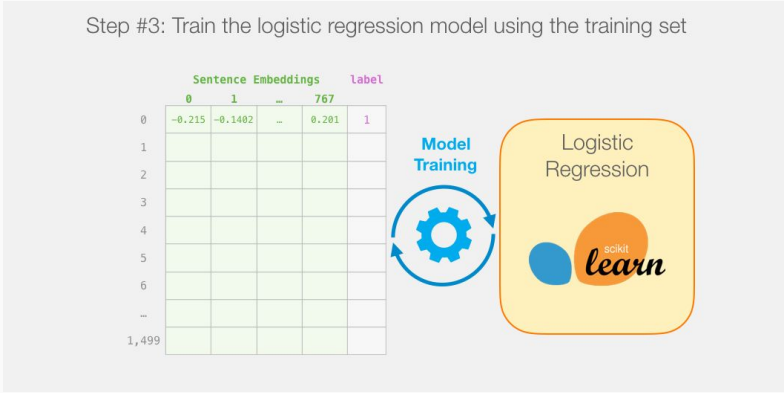
$y = Wh + b$

This means the internal representation space learned during pretraining remains unchanged, and the downstream task relies entirely on the separability of classes in that existing space.

Feature Extraction: Freezing the Pretrained Model







Source: https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Pretrained transformer models are trained on extremely large corpora and learn general linguistic abstractions, including:

- Syntactic structure
- Core semantic relations
- Context-sensitive disambiguation
- Common discourse patterns

Because of this, many downstream tasks are already partially encoded in the representation space.

In many cases, the task reduces to learning a linear decision boundary over the pretrained embeddings.

This phenomenon is often studied through linear probing, where a simple classifier is trained on frozen representations to measure how much task-relevant information is already encoded.

Does Feature Extraction Often Work?

- **But what if your task requires subtle distinctions not learned during pretraining?**

Examples:

- Sarcasm detection
- Domain-specific toxicity
- Biomedical classification

Does Feature Extraction Often Work?

Fine-tuning is a transfer learning strategy in which all (or most) parameters of the pretrained transformer are updated using task-specific supervision.

Instead of keeping θ fixed, we optimize:

$$\theta, W, b$$

under the new objective:

$$L_{task}$$

This allows the internal representation space to adapt to the statistical structure of the downstream dataset.

Fine-tuning does not merely adjust the decision boundary — it reshapes the embedding geometry itself.

Fine-Tuning: Updating the Entire Model

In fine-tuning, gradients flow through the entire network. This means updating:

- Self-attention weights
- Query/Key/Value matrices
- Feedforward layers
- LayerNorm parameters

The internal space rotates, stretches, compresses to make classes more separable.

Feature extraction → adjust boundary

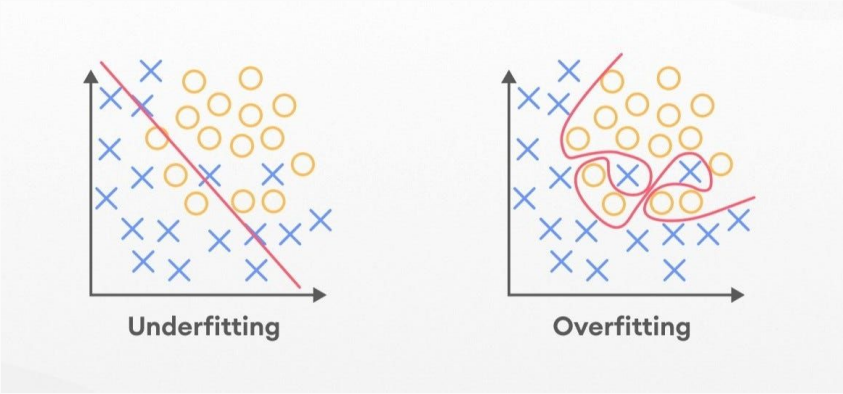
Fine-tuning → adjust space

What are the risks of adjusting the entire space?

Fine-Tuning: Updating the Entire Model

From a geometric perspective, pretrained models define a high-dimensional representation space in which inputs are embedded.

- Feature extraction:
  - Keeps the embedding space fixed
  - Learns a linear separator in that space
- Fine-tuning:
  - Modifies the embedding space itself
  - Changes distances between points
  - Alters cluster structure
  - Adjusts attention patterns



This process effectively specializes the model toward the downstream task distribution.

Representation Geometry: What Actually Changes?

Transformer models such as BERT-base contain over 100 million parameters.

When fine-tuning on small datasets (e.g., a few thousand examples), there is a severe parameter-to-sample imbalance.

This creates a high-variance learning regime where the model may:

- Memorize training examples
- Overfit spurious patterns
- Fail to generalize to new inputs

This problem becomes more pronounced as model size increases.

What strategies reduce overfitting? (Task)

Overfitting and Scale Mismatch

Catastrophic Forgetting in Fine-Tuning

Catastrophic forgetting refers to the phenomenon in which a pretrained model loses previously acquired general knowledge while adapting to a specific downstream task.

During fine-tuning, gradients update all parameters to minimize the task-specific loss.

However, this optimization does not explicitly preserve the information learned during pretraining.

As a result, the model may:

- Over-specialize to the downstream dataset
- Lose general linguistic knowledge
- Perform worse on broader language understanding tasks

This risk increases when:

- The downstream dataset is small
- The learning rate is high
- The domain is narrow

Catastrophic Forgetting in Fine-Tuning

Catastrophic forgetting refers to the phenomenon in which a pretrained model loses previously acquired general knowledge while adapting to a specific downstream task.

During fine-tuning, gradients update all parameters to minimize the task-specific loss.

However, this optimization does not explicitly preserve the information learned during pretraining.

As a result, the model may:

- Over-specialize to the downstream dataset
- Lose general linguistic knowledge
- Perform worse on broader language understanding tasks

This risk increases when:

- The downstream dataset is small
- The learning rate is high
- The domain is narrow

Not all transformer layers encode the same type of information.

- Lower layers capture syntactic and surface-level patterns
- Middle layers capture compositional semantics
- Higher layers are more task-sensitive

Therefore, one strategy is to:

- Freeze lower layers
- Fine-tune only higher layers

This approach aims to:

- Preserve general linguistic structure
- Reduce overfitting
- Lower computational cost

It represents a compromise between feature extraction and full fine-tuning.

Partial Fine-Tuning and Layer Freezing

Bias in Pretrained Models

Pretrained language models learn statistical patterns from massive text corpora.

These corpora reflect:

- Cultural norms
- Social stereotypes
- Historical inequalities
- Demographic imbalances

Because models learn from distributional patterns, they inevitably encode biases present in the data.

Examples of bias may include:

- Gender associations with professions
- Racial or cultural stereotypes
- Toxicity misclassification across dialects

Bias is not explicitly programmed — it emerges from large-scale statistical learning.

“Models reflect data. If data reflects society, models reflect society.”

If “doctor” co-occurs more frequently with male pronouns, embedding space may associate doctor with male features.

Fine-tuning can amplify bias under certain conditions. If the downstream dataset:

- Is small
- Is demographically skewed
- Contains annotation bias
- Has class imbalance

Then optimization may reinforce spurious correlations. For example:

If 80% of “angry” labeled examples come from one demographic group, the model may associate that demographic with anger.

Fine-tuning adapts the representation space to match dataset statistics —even when those statistics reflect imbalance rather than truth.

Bias Amplification During Fine-Tuning

- Basic implementation for fine tuning vs feature extraction
- Experimental comparison for Frozen vs Fine tuned
- Error analysis and Evaluation

-Tasks

In the next class ...