

CODECON Paper Reading Series: “Deep Learning for Economists” by Melissa Dell

Melissa Dell

Aug 2024

Takeaways from this paper

- ▶ A non-technical introduction to mainstream deep learning methods, friendly for people with no background knowledge.
- ▶ Recommendations to many excellent papers for people who want learn more.
- ▶ A guidance to help choose which method should be used.

What and why is deep learning

Deep Learning (DL) is an approach for learning representations of data from empirical examples. These representations simplify high dimensional unstructured data into continuous vectors.

- ▶ Exposure to massive data, and don't just learn from the problem in hand.
- ▶ Efficient in generating contextualized representations.
- ▶ The representation can be used for other downstream tasks with fine-tuning.
- ▶ Reduce manual feat

Classification

Many applications can be fall under the umbrella of *classification*. For example, textul sentiment analysis, textual topics identification, information extraction.

Here is a summary for possible applications:

TABLE 1—APPLICATIONS

Problem	Modality	Application(s)	Section
Classifiers and GenAI			
Sequence classification	Text	Classify news article topics	VI.3
Token classification	Text	Tag people, locations, orgs	VI.4
Paired text classification	Text	Text b entails a?	VI.5
Embedding Models			
Link structured data	Text, Images	Link firms, products, locations	VII.2
Link unstructured data	Text	Link people mentions to Wikipedia	VII.3
Classification w/ unknown categories	Text Images	Track content dissemination; data exploration	VII.4
Retrieval	Images	Optical character recognition	VII.5
Regression			
Object detection	Images	Detect document layouts	VIII

Note: Applications covered in this review.

Classification

Here is a flow chart to select methods for approaching classification:

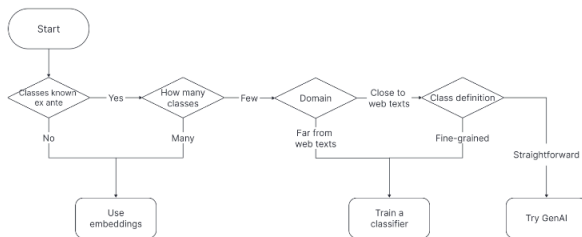


FIGURE 1. FLOWCHART FOR APPROACHING CLASSIFICATION.

Figure: Flowchart for Classification Tasks

I will use a NLP case to illustrate it in detail.

Classification

Carry out a classification task on a text dataset (annual reports):
If the task is:

- Sentiment analysis

STEP1 the classes are known and few (positive OR negative)

STEP2 it is close to web texts (there have been many labeled datasets on texts and emotion). Positive and negative emotion are easy to distinguish

STEP3 training a classifier OR GenAI are both proper (actually, training a classifier is better, which will be explained later)

- Identify digital transformation (DT) related expressions

STEP1 the classes are known (DT-related or not)

STEP2 many web texts are about it today, but there is no clear definition and no enough labeled data to clarify it.

STEP3 train a classifier is the safest way, but GenAI is also worth to try.

Foundational Deep Learning Architectures

1. Basics of Neural Networks

Activate functions: introduce non-linearity, enable the network to capture non-linear relationships.

Backpropagation: weights are adjusted using gradient descent (chain rule)

- ▶ **Convolutional Neural Network (CNNs)**: still widely used in computer vision tasks. Convolutional layers are the core building blocks of a CNN. The layer's parameters consist of a set of learnable filters. These filters are only applied to the nodes immediately surrounding a given node when computing the output for the next layer.

Foundational Deep Learning Architectures

- ▶ **Recurrent Neural Networks:** historically played an important role in NLP. RNNs process a sequence of inputs iteratively. At each time step, they maintain a state that captures historical information about the input sequence.
- ▶ **Transformer:** All tokens in a sequence are fed into the model in parallel and the model attends to all other tokens in the context to create contextualized representation for each token.
 - ▶ GPT: Generative (decoder) models: learn from prior tokens and predict the next one.
 - ▶ BERT: masked (encoder) language models: create contextualized representations of words in the sequence by learning bidirectionally.

Foundational Deep Learning Architectures

Optimizing Neural Networks

One experience is that when performance is unexpectedly poor, it is most often due to either a poorly chosen learning rate or incorrectly formatted input data.

Training Data

When conducting supervised learning, labeled data are further divided into training data and validation data. The main challenge is to find a validation dataset which is representative to all other data.

- ▶ Random sampling: not applicable when classes that the researcher would like to measure are highly imbalanced. For example, articles about a topic of interest appears only once in every ten thousand texts. The labelling requirements for sampling enough positives are infeasible.
- ▶ Selecting using keywords is biased.
- ▶ Embedding models: measure the similarity between texts and queries about a class (e.g., “this article is about tax policy”). The higher the similarity, the more likely it comes from that class. It can also provide informative negatives for training.

Classifiers

A neural network predicts a score for each of N classes and the input is assigned the class with the highest score.

Notices when conducting:

- ▶ When creating labels, the labeled data should be relatively balanced across classes.
- ▶ Loss functions: Support Vector Machine (SVM) loss and cross-entropy (CE) are two most commonly used loss functions.
- ▶ F1 score: an evaluation on the performance of the prediction model. If either the precision or the recall is low, the F1 score will also be low.

Generative AI for classification

Large language models like GPT, Claude, or Llama can be used to label text with proper prompts.

Like a blackbox. A few clear insights:

- ▶ Prompt tuning should be done on a validation set. The performance on test set speaks nothing.
- ▶ Breaking task down into simple steps. Simple prompts work much better than lengthier and more detailed ones.
- ▶ The comparison among LLMs in this paper shows that GPT-4o outperforms than other models (tested only on text in English, our tests show that some China's model perform better on texts in Chinese).

Generative AI for classification

Advantages:

- ▶ startup costs are low
- ▶ it can be used zero-shot (without the user providing training data), whereas training a classifier requires training data.

Disadvantages:

- ▶ Does not provide the same fine-grained control as training a classifier (provide more data to improve performance, for example)
- ▶ Interpretability and reproducibility: commercial API is not reproducible if its corresponding deprecated.

How to choose? Create test and validation sets first. Try GAI first and decide to train a model or not depending on the data you have.

Tuned BERT for classification

The classifiers were trained with LinkTransformer, which supports using any base language model available on Hugging Face.

RoBERTa, DistilRoBERTa are widely used, improved version of BERT.

In most cases—across a diversity of topics—the tuned classifier tends to outperform or equal the performance of GPT.

Compared to sparse method (e.g., TF-IDF), in which each term in the corpus forms a dimension in the vector space. Sparse methods are useful when exact term overlap is highly informative. But relying on term overlap is often a major shortcoming. Neural methods address these shortcomings to map texts to a dense vector representation.

Token Classification

For example, extract name from sentences (Named Entity Recognition, NER).

Generative AI can also well handle such problems.

Relationships between texts

Measure whether two text are related in some pre-specified way.
For example, we would like to classify whether one statement entails another: do they take the same stance on a political issue?
Does one follow the other?

Two approaches to comparing texts

- ▶ cross-encoder: put them together then embed them and make comparison. This allows full cross-attention between all tokens, but requires much computational resources.
- ▶ bi-encoder: embed them separately then make comparison. This is the most commonly used method.

Relationships between texts

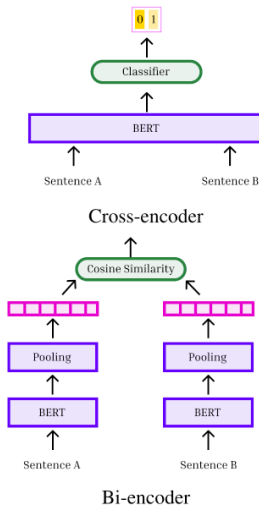


Figure: Architectures to compare texts

Embedding Models

When the classes are no known ex ante or the number of classes is too large, it is efficient to work with the embeddings from the final layer of the transformer or CNN.

Five applications will be discussed:

- ▶ Calculate the semantic similarity
- ▶ Record linkage with structured data
- ▶ Linking unstructured data
- ▶ Classification when categories are unknown
- ▶ Optical character recognition

Application1: Calculate the semantic similarity

Working directly with embeddings requires distances between vector representations to be meaningful. The geometric properties of pre-trained transformer language models are not well-suited to this task, because representations of low-frequency words are pushed outwards on the hypersphere. Mathmatically, the roblem is that the embedding space created by a pre-trained trainsformer space is not *isotropic*.

Contrastive Learning is a widely used method to improve the isotropy.

Application1: Calculate the semantic similarity

The contrastive loss function encourages the model to reduce the distance in embedding space between positive examples (e.g., similar texts or images) and increase the distance between negative examples (e.g., dissimilar texts or images)

Details about constastive learning

- ▶ Bi-encoder setup is used. The cosine distance then is used to compare the similarity.
- ▶ Many options for the loss functions, such as Cosine loss, InfoNCE loss.
- ▶ Selecting informative negative examples is important. They cannot be too 'easy'. Researchers should use prior knowledge to select 'hard' negatives for training. **One good way** to mine is to use a pre-trained model to choose negative examples with similar embeddings.

Application1: Calculate the semantic similarity

Fine-tuning will accentuate the relevant dimensions, creating better separation between classes in embedding space.

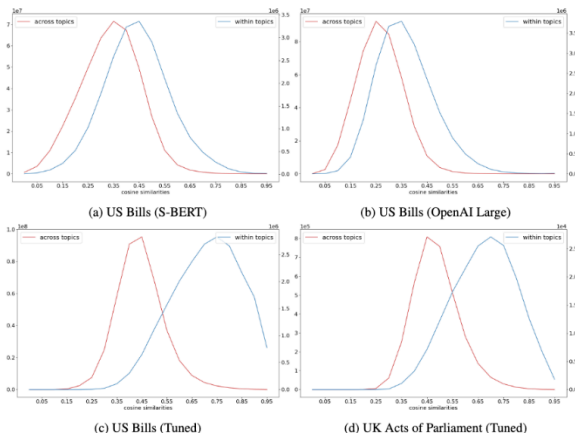


FIGURE 7. EMBEDDING SIMILARITIES WITHIN AND ACROSS TOPICS.

Figure: Enter Caption

Application2: Record linkage with structured data

A research might need to link individuals, locations, firms organizations across datasets. For example, linking modern firms and products across six languages.

LinkTransformer is a package for using transformer models for record linkage that is geared towards social scientists

<https://linktransformer.github.io/>.

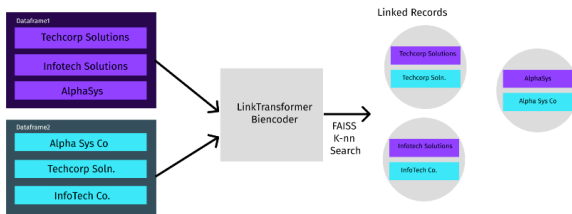



FIGURE 8. LINKTRANSFORMER ARCHITECTURE.

Figure: Enter Caption

Application2: Record linkage with structured data

More details about *LinkTransformer*

Introducing LinkTransformer 

Link data frames with 2-3 lines of intuitive code

CompanyName	Industry	Founded_Year
TechCorp	Technology	2005
InfoTech Solutions	Technology	1998
GlobalSoft Inc	Software	2010
DataTech Co	Data Analytics	2012
SoftSys Ltd	Software	2003
TechCorp	Technology	2005

Merge on
Company Name

CompanyName	Revenue (Millions USD)	Num_Employees	Country
Tech Corporation	5000	10000	USA
InfoTech Sole	4500	8500	Canada
GlobalSoft Incorporated	3000	6000	India
DataTech Corporation	2500	5000	Germany
SoftSys Limited	4000	7500	UK
TechCorp	5500	12000	USA
AlphaSoft Systems	3800	7000	Spain

```
import linktransformer as lt

df_lm_matched = lt.merge(df2, df1, merge_type='1:n', on="CompanyName", model="all-MiniLM-L6-v2",
left_on=None, right_on=None)
```

Figure: Easy to use in LinkTransformer

- ▶ Allows user to employ Sentence Transformer models
- ▶ Provide APIs to use LLM for other data processing tasks e.g. classification, aggregation, de-duplication.

OCR helps to link text and picture.

Application3: Record linkage with unstructured data

Entity disambiguation: Linking entity mentions in raw texts to Wikipedia or other knowledge bases to disambiguate information

Entity mentions are disambiguated by embedding their contexts with the disambiguation model and retrieving their nearest Wikipedia neighbor in embedding space. If they are below a threshold cosine similarity to the nearest Wikipedia embedding, they are marked as not in the knowledge base.

Application4: Classification when categories are unknown

Detecting reproduced article texts and images, classifying the biggest news stories historically.

Unsupervised clustering. For example, based on a historical U.S. newspaper articles, determine the biggest news stories of each year without knowing what these stories are ex ante.

- ▶ A bi-encoder embedding model constructively trained is useful.

Application5: Optical character recognition (OCR)

An important task to recognize texts from pictures, especially for economic historians.

The problem is the OCR quality of existing off-the-shelf solution is always poor.

- ▶ *EffOCR* is an OCR architecture designed for researchers, which can be customized by contrast learning.
- ▶ EffOCR performs very accurately, even when using lightweight models designed for mobile phones that are cheap to train and deploy.

Regression

Regression is analogous to classification, except that a regression layer added to a neural network predicts a continuous number(s). One application is on object detection. For example, an economist wishing to measure informality from street view data would need to localize street vendors in an image.

- ▶ Off-the-shelf solution works well.

Alternative methods

There are also other alternative methods for the applications discussed before. Generally speaking, they work worse than the methods mentioned before.

- ▶ OCR: *seq2seq* vs. *EffOCR*
- ▶ Entity disambiguation: *LUKE*, *GENRE*