Deep Learning for Economics

Melissa Dell

October 2024

The Deep Learning Revolution

- Deep learning has revolutionized the processing of *unstructured* data (text, images, audio, video)
- This has in turn transformed a variety of disciplines, ranging from allowing NASA to land a rover on rugged Martian terrain to changing how doctors diagnose disease
- Deep learning can similarly be used to transform economic analyses

Unstructured Data in Economics

Massive quantities of non-computable data could power economic analyses if converted into a computable format:

- Text contains massive amounts of unstructured information
- Data can be trapped in images; also audio and video
- Libraries and archives have scanned billions of pages of historical documents
- While the raw information is very different, DL methods to convert them into computable information are quite related, often drawing on the same neural net architecture

Two Approaches to Unstructured Data

Generally speaking, there are two distinct approaches to processing unstructured data:

- **Rule-based approach:** Write a set of instructions that tells the computer how to process the data

- **Deep learning:** Learn how to process the data from empirical examples

Deep Learning

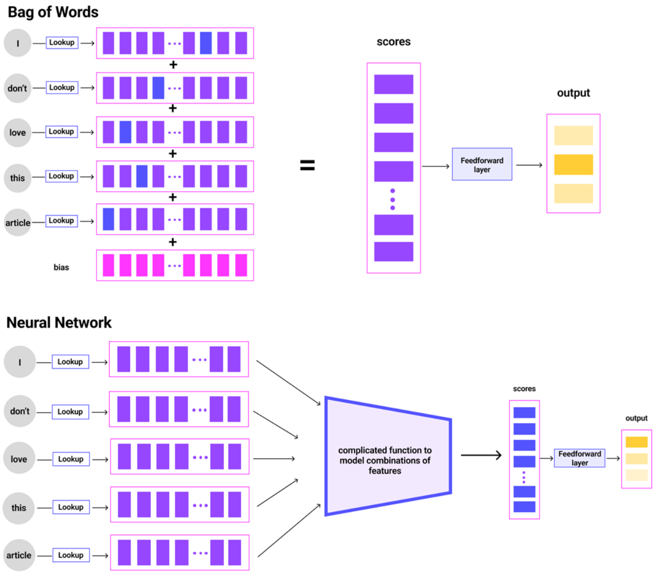
- Deep neural networks map typically unstructured data - such as text, document image scans, satellite and other imagery, videos, and audio - to a continuous vector space
- In other words, they map complex and diverse types of data into a format that is easier to process and understand
- At its core, deep learning is an approach for learning representations of data from empirical examples (LeCun, Bengio and Hinton, 2015)

Why Deep Learning

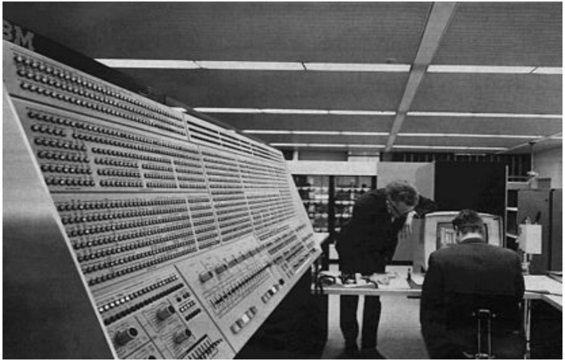
Why would one use a neural network to transform raw data into vector representations, versus just working directly with raw texts or images?

- **Transfer learning:** Deep neural nets incorporate relevant information in their parameters from exposure to massive-scale data
- **Context:** Raw pixels or words lack context. Deep neural networks provide a powerful method for computing contextualized representations
- **Scalable:** Raw texts and images are computationally unwieldy. In contrast, there are extremely optimized tools for continuous vector computations

Context


Bag of Words
I
don't
love
this
article
Lookup
bias
scores
Feedforward layer
output
Neural Network
complicated function to model combinations of features
scores
Feedforward layer
output

Computing in Economics



While computing has a long history in economics, the advent of personal computing in the 1990s revolutionized the discipline. Today, advances in GPU compute and the availability of cheap cloud compute again have the potential to transform the types of data and questions economists can study

Deep Learning for Prediction

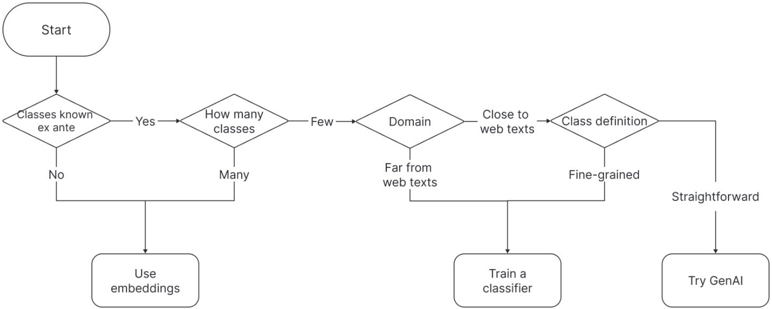
- Neural networks excel at imputing low-dimensional structured data from unstructured texts or images

- Conceptually, we can divide problems into:
    - Imputing a continuous number (regression)
    - Imputing a pre-specified discrete class (classification)
    - Imputing relationships in data when the classes are not specified ex ante

Deep Learning for Prediction

- A neural network encodes unstructured data into lower-dimensional vectors
- The researcher can use these representations to predict whether the raw data belong to pre-specified class(es) by adding a classifier layer; regression works analogously
- Broadly speaking, generative AI performs classification by predicting what word (in a pre-specified vocabulary) comes next
- Alternatively, one can work directly with the vector representations, referred to as *embeddings*

Classification Flow Chart



Embedding models

- Embeddings can be used to group like data together, typically through clustering or KNN-retrieval
- Off-the-shelf transformer (e.g., BERT, RoBERTa, etc.) embeddings have undesirable geometric properties (Ethayarajh, 2019)
- A method called contrastive training can be used to make the distances between embeddings in the vector spaced created by a neural network more meaningful (Wang, 2021)

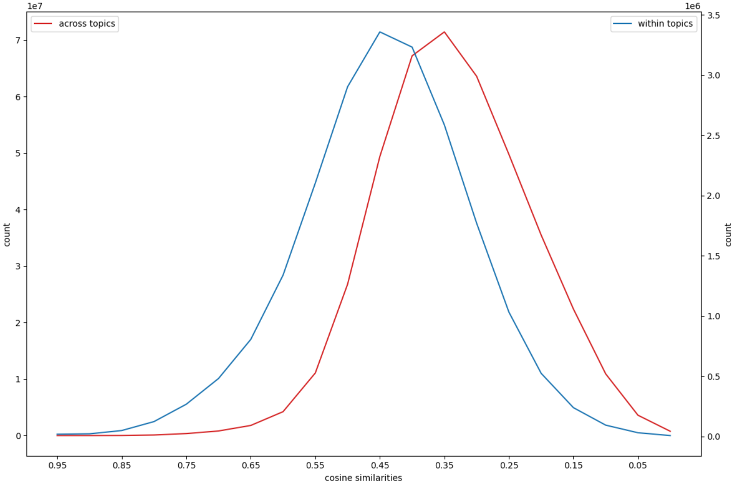
Contrastively trained embedding models

- Contrastively trained embedding models learn a mapping from unstructured data to continuous vector space, such that instances that belong to the same class have similar embeddings and instances that belong to different classes have dissimilar embeddings
- More details on contrastive learning are given in my JEL review article “Deep Learning for Economists”, or on the contrastive learning post at https://econdl.github.io/
- There is a lot of different information incorporated in an off-the-shelf embedding (e.g., Rogers et al., 2020)
- Hence, some fine-tuning is often necessary for the model to learn what dimension(s) of texts/images are of relevance to the task at hand

Contrastively trained embedding models

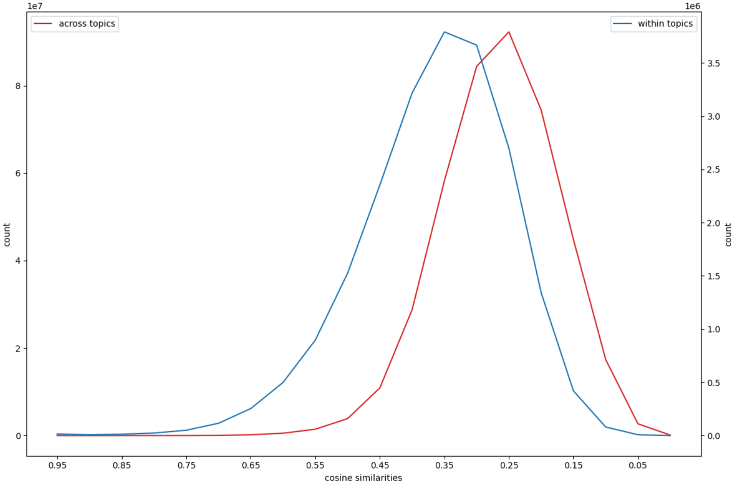
- Consider the Comparative Agendas dataset, which has high quality topic tags about legislative acts
- Off-the-shelf embeddings from Sentence Bert and OpenAI do not do a particularly good job of grouping legislation on different topics together
- However, with limited paired training data, we can contrastively tune a model to map legislation on different topics to different regions of embedding space, and legislation on the same topic nearby
- This model translates well across legislative settings

Off-the-shelf (S-BERT) embedding model and U.S. congressional bills data



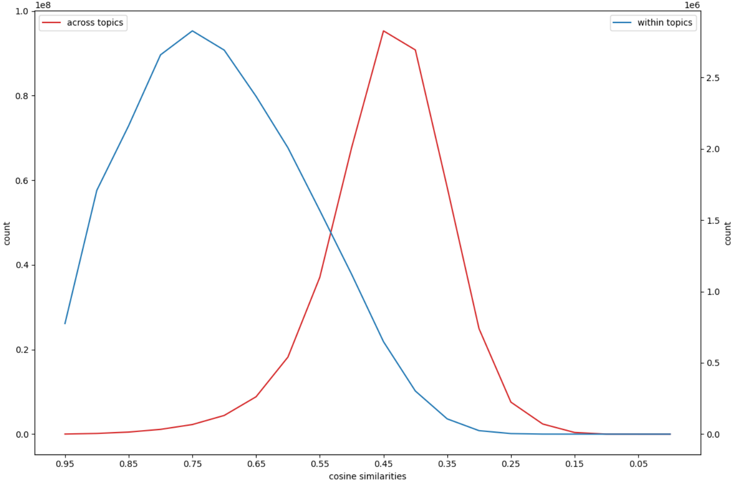
Cosine similarities between the embeddings of legislative texts, within topic (blue) and across topic (red)

Off-the-shelf (OpenAI) embedding model and U.S. congressional bills data



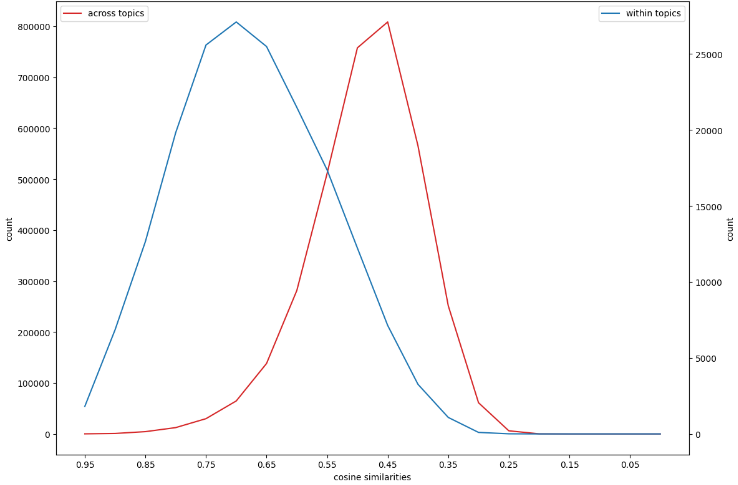
Cosine similarities between the embeddings of legislative texts, within topic (blue) and across topic (red)

Fine-tuned embedding model and U.S. congressional bills data



Cosine similarities between the embeddings of legislative texts, within topic (blue) and across topic (red)

Fine-tuned embedding model and UK Parliamentary Acts data



Cosine similarities between the embeddings of legislative texts, within topic (blue) and across topic (red)

Applications of Embedding Models

- Data exploration (classes unknown ex ante)
    - Reproduction of news articles
    - Biggest stories
    - Most similar news stories to a query
- Many classes
    - Record linkage in structured data
    - Linking individuals mentioned in unstructured texts
- Adding classes at inference time
    - Document transcription

Detecting Reproduced Content with Embeddings

- Detecting noisily reproduced content is important both for first order social science questions and for commercial applications
    - Media historian Julia Guarneri (2017) writes: "by the 1910s and 1920s, most of the articles that Americans read in their local papers had either been bought or sold on the national news market... This constructed a broadly understood American 'way of life' that would become a touchstone of U.S. domestic politics and international relations throughout the twentieth century."
    - Important for de-duplicating training data and controlling test set leakage

Reproduced Content

- We develop novel methods - combining a customized embedding model and single linkage clustering - for detecting noisily reproduced content (Silcock, D'Amico-Wong, Yang, and Dell, 2022)

- We have applied this to detecting noisily reproduced historical news articles on a massive scale (Silcock, Arora, D'Amico-Wong, and Dell, 2024) and to evaluating test set leakage in foundation models (Sainz et al., 2024)

Newswire Dataset

Description	Count
Front page articles	137,941,190
Unique articles (including singletons)	99,472,910
Unique articles reproduced > 3 times	2,889,012
Unique wire articles	2,719,607
Total reproductions of wire articles	32,107,676

Table: Counts of articles meeting various criteria in our raw digitized newspaper corpus.

Domestic Datelines

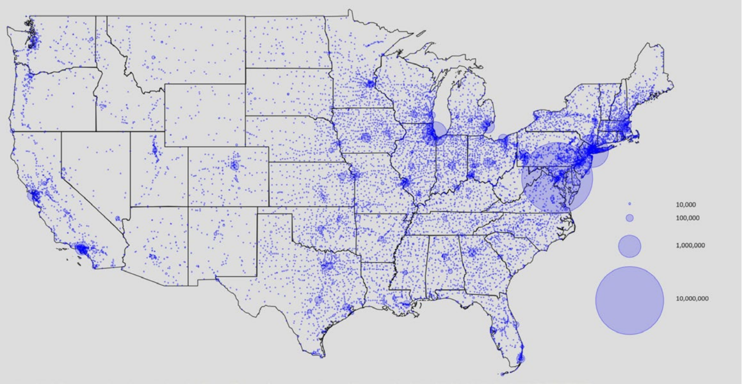


Figure: Reproduction of newswire articles with domestic datelines.

International Datelines

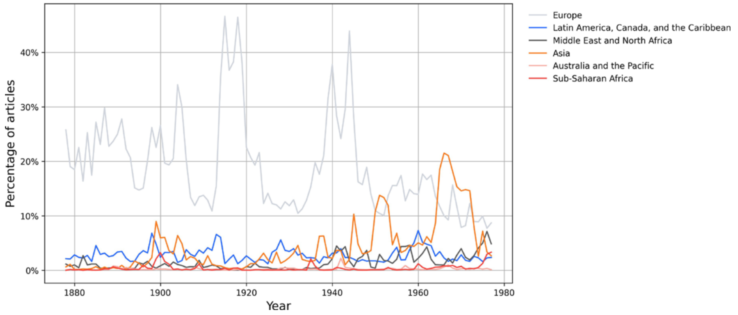


Figure: Reproduction of newswire articles with international datelines.

Newswire Topics

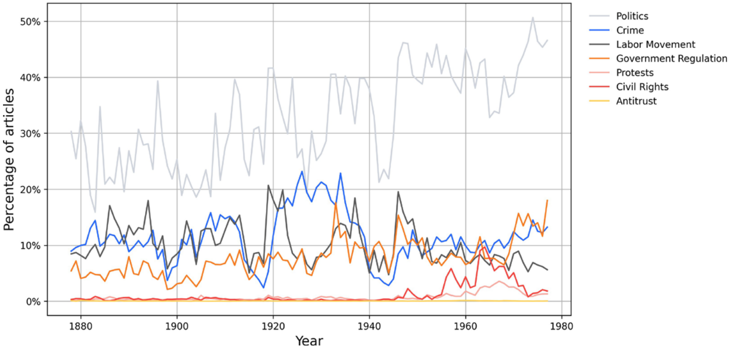


Figure: Share of reproduced newswire articles with a given binary topic tag, across time.

Newswire Topics

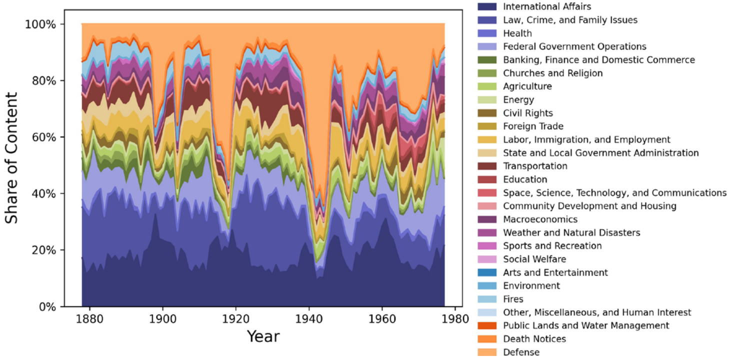


Figure: The distribution of multiclass topic tags, trained on data from the Comparative Agendas project.

Computational Advantages

	Embed Articles	Compute Similarity	Build Graph	Commun. Detect.	Total Time	Mean Times Reproduced
Bi-Encoder	8:38:52 (GPU)	3:04:53 (GPU)	0:01:23 (GPU)	0:00:02 (GPU)	11:45:10 (GPU)	6.41
Hashing		3:39:05 (CPU)	0:00:55 (GPU)	0:00:08 (GPU)	3:40:08 (mostly CPU)	11.55

Table: This table reports computational efficiency in scaling the bi-encoder and LSH methods to a 10 million article corpus. Parentheses indicate whether the calculations were run on a CPU or a single NVIDIA A6000 GPU card. *Mean times reproduced* reports the average size of duplicated article communities that each method estimates.

Biggest Stories of the Year

- We would like to know which news stories receive the most coverage
- We have no idea what those stories are ex ante - cannot use a classifier
- Instead, custom train a large language model to map articles about the same story to similar vector representations using paired data from allsides
- Cluster the resulting embeddings with single linkage clustering

American Stories Dataset

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Total	Text Bounding Boxes				Other Bounding Boxes			
	Boxes	Articles	Headlines	Captions	Bylines	Images	Ads	Tables	Mastheads
Legible	-	335M	368M	9.7M	14.7M	-	-	-	-
Illegible	-	26M	27M	0.9M	2.5M	-	-	-	-
Borderline	-	77M	22M	1.3M	1.2M	-	-	-	-
Total	1.14B	438M	417M	11.9M	18.4M	9.1M	221M	16.3M	4.9M

Table: Dell et al. (2023)

Biggest Stories

Year	Biggest story	Year	Biggest story
1885	Death of General Grant	1903	Panama Canal Treaty
1886	Southwest Railroad Strike	1904	Russo-Japanese War
1887	Vatican supports Knights of Labor	1905	Russo-Japanese Peace Process
1888	Rail strikes	1906	Hepburn Railroad Rate Bill
1889	Samoan Crisis	1907	Mining accidents
1890	1893 World's Fair planning	1908	Taft presidential victory
1891	New Orleans Lynchings	1909	Race to the North Pole
1892	Homestead Steel Strike	1910	Rail strikes
1893	World's Fair, Chicago	1911	Canadian Reciprocity Bill
1894	Wilson–Gorman Tariff Act	1912	Republican National Convention (Taft v Roosevelt)
1895	British occupation of Corinto, Nicaragua	1913	Underwood-Simmons Tariff Act
1896	Bimetallism Movement	1914	World War I
1897	Coal Miners' Strike	1915	World War I
1898	Cuban War of Independence	1916	Pancho Villa Expedition
1899	Philippine-American War	1917	World War I
1900	Anglo-Boer War	1918	World War I
1901	U.S. Steel Recognition Strike	1919	Treaty of Versailles
1902	Anthracite Coal Strike	1920	Rail strikes

News Déjà Vu

We use the same model (after masking named entities) to find the most similar historical story to current news stories (Franklin, Silcock, Arora, Bryan, and Dell, 2024).

OPENAI LAYS OUT PLAN FOR DEALING WITH DANGERS OF AI

Gerrit De Vynck for The Washington Post

December 18, 2023

OpenAI the artificial intelligence company behind ChatGPT laid out its plans for staying ahead of what it thinks could be serious dangers of the tech it develops such as allowing bad actors to learn how to build chemical and biological weapons. OpenAI's "Preparedness" team led by MIT AI professor Aleksander Madry will hire AI researchers computer scientists national security experts and policy professionals. …

Read the full article **here**.

NEW COMPUTOR MAY MAKE PEOPLE OBSOLETE

News Wire Article published in the Somerset Daily American

March 30, 1950

NEW BRUNSWICK N. J. March 28—(4)—A new mechanical brain—resembling a pinball machine on a jackpot rampage—may make _ people obsolete The device described as capable of operating a complete factory without human aid is designated officially as the magnetic drum digital differential analyzer. Its inventor 31-year-old physicist Floyd Steele calls it Maddida for short. And what Maddida can do was shown .today at the opening of a three-day conference on computing jachinery at the Rutgers university coliege of engineering. Maddida is primarily a computing device

Record Linkage

- Linking information across diverse sources is fundamental to many pipelines.
- For example, researchers and businesses frequently link individuals or firms across censuses and company records or link products/industries across datasets

Traditional Record Linkage

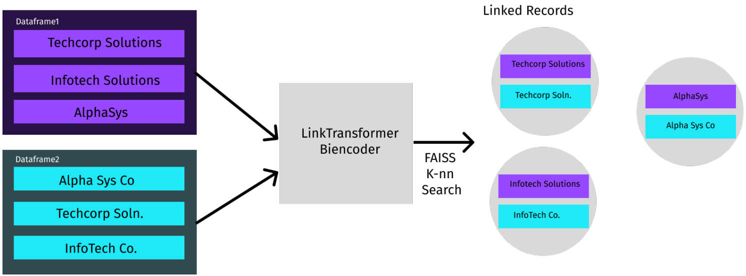
- Conventional string similarity measures rely on edit distance or n-gram overlap to determine how two strings differ - they are unable to bring in semantic information
- For example, the distance between ABC Co. and ABC Corporation shouldn't be high, as Co. is a common abbreviation of the word Corporation
- Yet these two string have a high Levenshtein edit distance.

LinkTransformer

A Unified Package for Scalable Record Linkage with Transformer Language Models

- LinkTransformer (Arora and Dell, 2024) brings large language models to data frame manipulation tasks like merges, deduplication, and clustering.
- It supports standard merging, merging with blocking and multiple keys, bypassing translation with cross-lingual merges, aggregation and classification, clustering, and de-duplication.
- It supports models on the Hugging Face Hub and OpenAI Embedding models. We've trained our own collection of over 20 open-source language models for 6 different languages and different tasks.

Embeddings for Record Linkage



LinkTransformer

A Unified Package for Scalable Record Linkage with Transformer Language Models

- The API is designed to be familiar to practitioners coming from environments like R and Stata.
- Training your own models is as easy as one line of code.
- It also includes a classification module, to facilitate sequence-level text classification

Basic Functionality

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 Soln	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

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

Multilingual Merging

Libellé du Produit	Catégorie	Prix
Ordinateur portable performant	Électronique	1400
Smartphone avec prise en charge de l'IA	Électronique	950
Logiciel CRM pour entreprises	Logiciel	1800
Plateforme d'analyse de données pour gros ense...	Analyse de données	1700
Gestion de documents pour les entreprises	Logiciel	2000

Merge on Product Name in French and English

Product Label	Category	Price
High-performance laptop	Electronics	1400
Smartphone with AI support	Electronics	950
CRM software for businesses	Software	1800
Data analytics platform for large datasets	Data Analytics	1700
Document management for businesses	Software	2000
High-resolution security camera	Electronics	1200

```
df_lm_matched = lt.merge(df_french, df_english, merge_type='1:m', left_on="Libellé du Produit",
right_on="Product Label", model="distiluse-base-multilingual-cased-v1")
```

Deduplication

CompanyName	Revenue (Millions USD)
Tech Corporation	5000
InfoTech Soln	4500
GlobalSoft Incorporated	3000
DataTech Corporation	2500
SoftSys Limited	4000
TechCorp	5500
AlphaSoft Systems	3800

Deduplicate on CompanyName →

CompanyName	Revenue (Millions USD)
Tech Corporation	5000
InfoTech Soln	4500
GlobalSoft Incorporated	3000
DataTech Corporation	2500
SoftSys Limited	4000
AlphaSoft Systems	3800

```
df_dedup=lt.dedup_rows(df,on="CompanyName",model="sentence-transformers/all-MiniLM-L6-v2",cluster_type=
 "agglomerative",
    cluster_params= {'threshold': 0.7})
```

Aggregation

Fine Category ID	Fine Category Name	Coarse Category ID
101	Laptops	1
102	Desktop Computers	1
103	Smartphones	1
104	Tablets	1
105	Headphones	1
106	Speakers	1
107	Refrigerators	2
108	Washing Machines	2

Merge from Fine to coarse product category

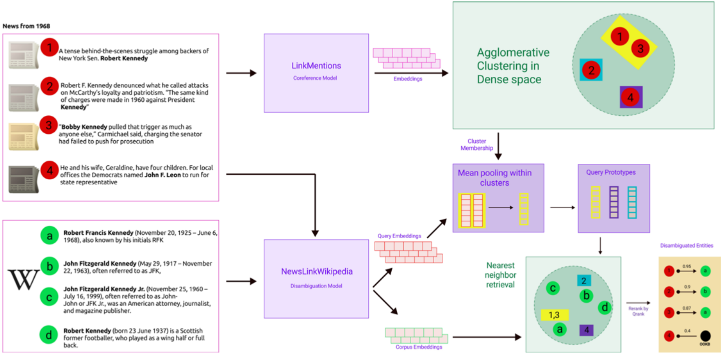
Coarse Category ID	Coarse Category Name
1	Computers & Electronics
2	Home Appliances
3	Clothing & Apparel
4	Sports & Outdoor

```
df_lm_aggregate = lt.aggregate_rows(df_fine, df_coarse, model="all-mpnet-base-v2", left_on="Fine
Category Name", right_on="Coarse Category Name")
```

Model Training

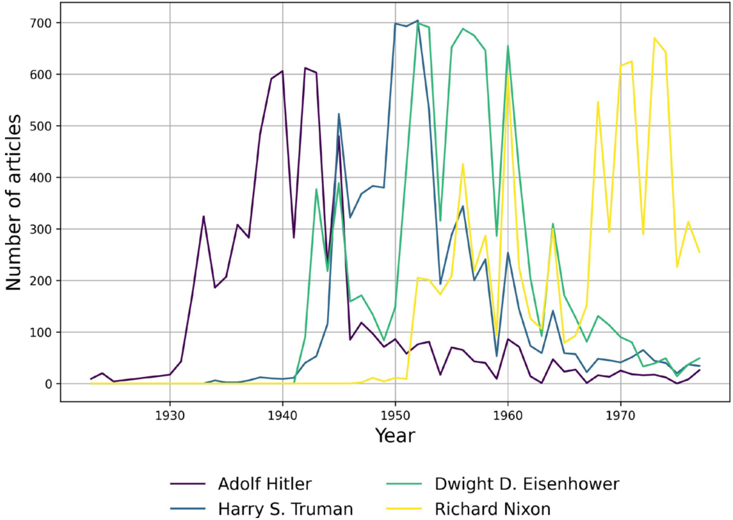
```
best_model_path=lt.train_model(
        model_path="hiiamsid/sentence_similarity_spanish_es", #A spanish sentence transformer model!
        data=os.path.join(lt.DATA_DIR_PATH, "es_mexican_products.xlsx"),
        left_col_names=["description47"],
        right_col_names=['description48'],
        left_id_name=['tariffcode47'],
        right_id_name=['tariffcode48'],
        log_wandb=False,
        training_args={"num_epochs": 1}
    )
```

Entity disambiguation with embeddings



Arora, Silcock, Heldring and Dell (2024).

Entity disambiguation

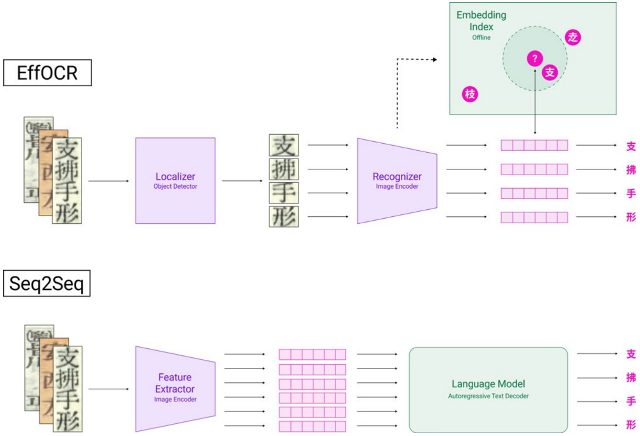


Mentions over time of entities that appeared most commonly in newswire articles.

Adding Classes at Inference Time

- Embedding models are also well suited to contexts where you may need to add additional classes after training the model, which cannot be done when using a classifier head
- One context where I frequently encounter this is transcribing documents

EffOCR Architecture



Carlson, Bryan, and Dell (2024)

EfficientOCR

- Our EffOCR package (Bryan, Carlson, Arora, and Dell, 2024) makes it straightforward to tune your own custom OCR model, including for very low resource settings
- We have a demo notebook that you can use to train your own OCR model to recognize polytonic (ancient) Greek
- We show that this customized model, which can be trained with free student compute credits, beats Google Cloud Vision, the state-of-the-art for ancient Greek

Conclusion

- Deep learning provides powerful tools for processing unstructured economic data, creating robust representations for downstream analyses
- Becoming familiar with deep learning methods, how they apply to economics, and how they can be implemented and debugged can entail significant startup costs
- If you are interested in learning more, check out https://econdl.github.io/