adept ID

Job Categorization - Take Home

Chenfei Song

03/20/2024

Overview

Objectives

Assist labor market data inference with job classification results based on occupational categories (O*NET).

Challenge

Many providers of labor market data would perform job classification task with complicated, rules-based systems, which

- depends heavily on expertise and assumptions of human experts, which can introduce subjectivity, bias or unfairness into the classification process, especially for diverse or unconventional job titles.
- can be time-consuming and labor-intensive to define, verify, maintain and update the system
- has limited flexibility to adapt to new or evolving data patterns under dynamic market changes

Dataset

- 50k sample data parsed job postings (ID, Post date, job title, body text) and their associated O*NET classifications (ONET Name, ONET)

Solution

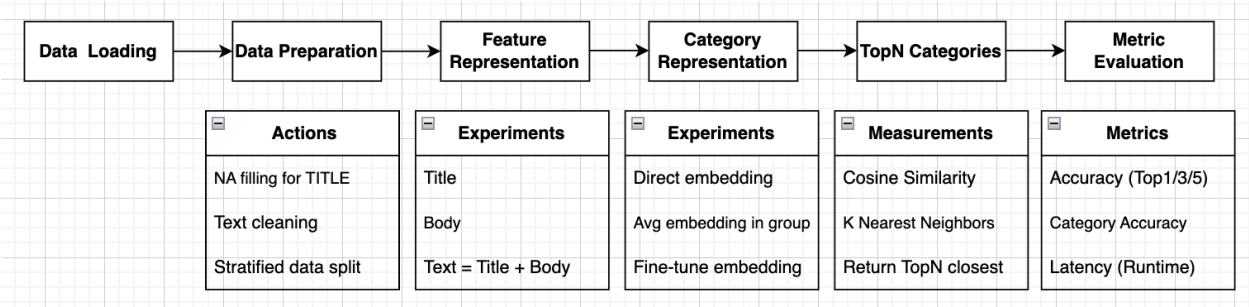
- Build an advanced classification model that ingests job postings and predicts topN most likely occupational categories based on a pre-trained language model.

Business use cases

- Advertise over disparate impact**:** find and support talent in new places and from diverse backgrounds for rare job categories
- Improve client's match score and end-customer engagement: accurate job category prediction could increase appended labor market data quality (expected wages and educational requirements)
- Provide better insights to US users outside of their target regions

Approach

Workflow



Key Assumptions

- Categories in the sample set are the full category set
- Categories are correctly labeled
- Rare categories are important for our business case (hard-to-fill roles & diversity)
- Job postings are english based

Fine-tuned category embedding

- **Fine-tune**
  - Group average embedding doesn't consider negative classes
- **Contrastive Loss**
  - Poor margin and lacking robustness of cross-entropy loss (800+ classes)
  - Contrastive loss focuses on hard negatives and inherently compare cosine similarities

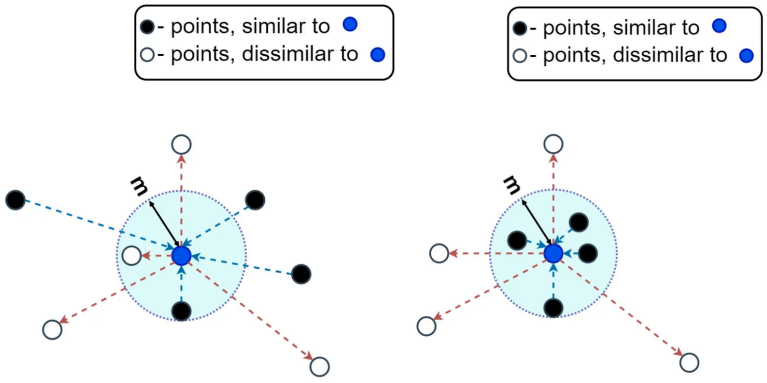


Figure 5 — What we would like the algorithm to do. Notice how the white dots that were outside weren't moved farther away from the margin.

$$(1-Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2$$

Metrics

- Performance-based
    - Overall accuracy (Top 1/3/5)
    - Accuracy by job category
- Engineering-based
    - Training time
    - Inference time

Analysis

The following and result performance is based on provided 50k samples, which is split into train set(32k), validation set (8k) and test set (10k), run on Google Colab T4 GPU.

Data Split with stratification

Among total 833 categories, 387 of them have less than 10 records; 263 less than 5, 114 with single record.

Feature Representation

- Title embedding has better accuracy than Body or Title+Body

Category Representation

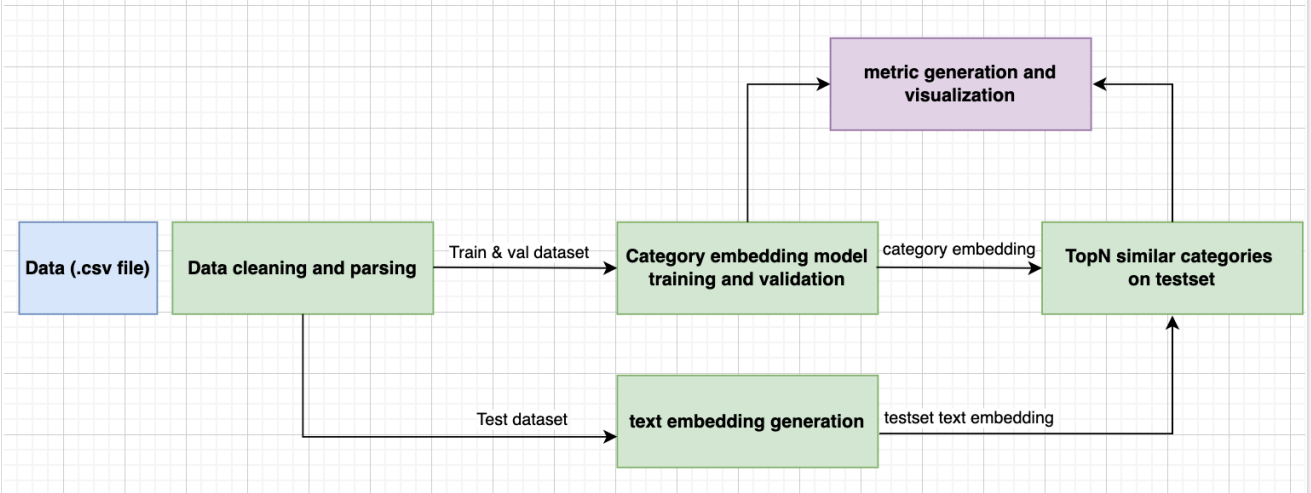
- Average title embeddings in the same category has better accuracy than direct embedding
- Fine-tuned category embeddings further improves accuracy, but requires 26X train time

	Top-1 Accuracy	Train & val Time (second)
Direct embedding	0.33	10s
Average group embedding	0.56	10s
Fine-tuned embedding	0.68	260s

Final Approach

- Use title feature to generate job posting embedding with pre-trained SBERT model
- Use Fine-tuned category embedding model to generate category embedding
- Final prediction is based on cosine similarity and KNN of title embedding and category embeddings

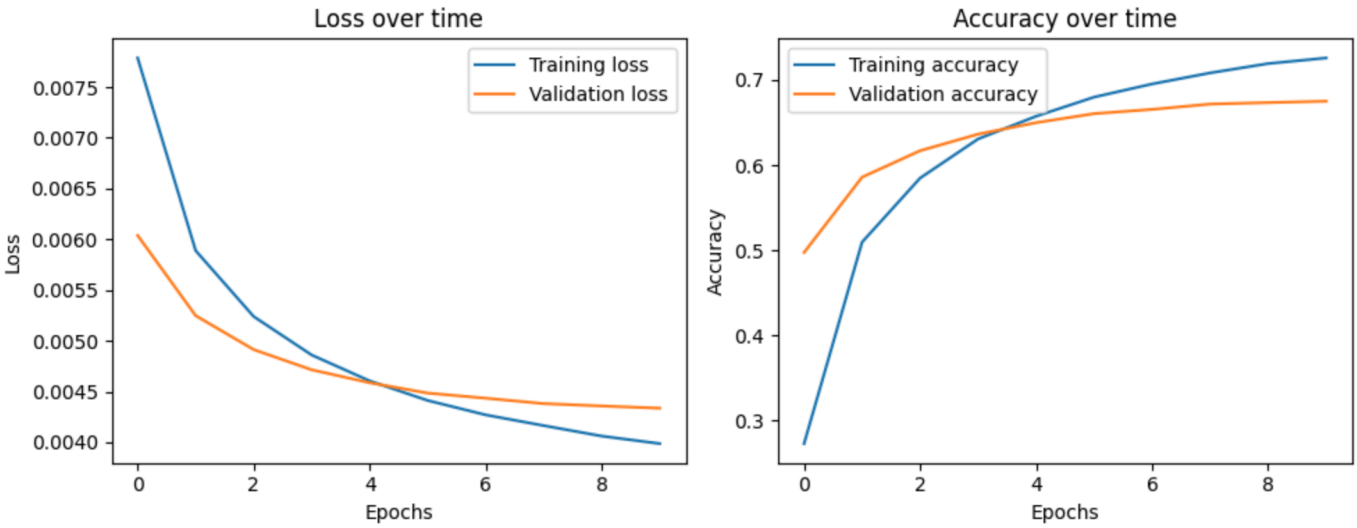
Code Architecture



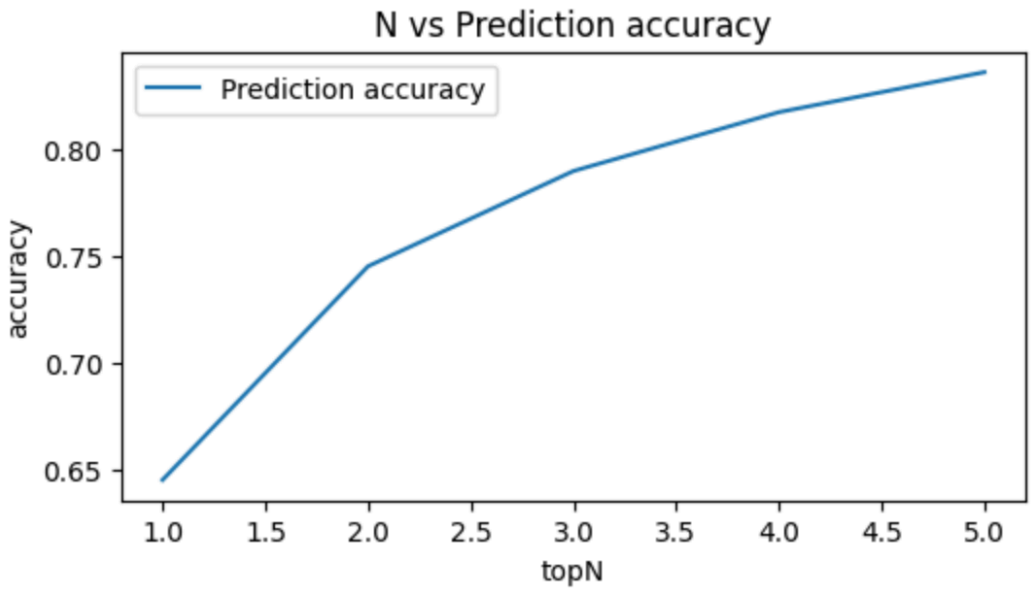
Result

Visualization

- Train & validation (Loss over time & Accuracy over time)



- Testing (N VS Prediction Accuracy)



Sample Test

```
---- Job Title: ------
Part-time RN - Medical/Surgical Nampa (flexible schedule)

---- ONET NAME: ------
Registered Nurses

---- Predicted ONET NAME: ------
['Registered Nurses', 'Medical Assistants', 'Critical Care Nurses', 'Medical and Health Services Managers', 'Nurse Practitioners']
```

Metrics

- **Train & validation**
    - **loss**: train (0.005); val (0.005)
    - **Top-1 accuracy**: train (0.66), val (0.65)
    - **Runtime**: 260s (30k+10k samples)

```
Epoch 1/5, Train Loss: 0.0078, Val Loss: 0.0060, Train Acc: 0.2652, Val Acc: 0.4945
Epoch 2/5, Train Loss: 0.0059, Val Loss: 0.0052, Train Acc: 0.5085, Val Acc: 0.5850
Epoch 3/5, Train Loss: 0.0052, Val Loss: 0.0049, Train Acc: 0.5872, Val Acc: 0.6165
Epoch 4/5, Train Loss: 0.0049, Val Loss: 0.0047, Train Acc: 0.6277, Val Acc: 0.6387
Epoch 5/5, Train Loss: 0.0046, Val Loss: 0.0046, Train Acc: 0.6594, Val Acc: 0.6509
```

- **Prediction**
    - **Top-1/3/5 Accuracy**: 0.64; 0.79; 0.84
    - **Runtime**: 9s (10k samples)

Next Steps

Depending on business priorities and resource optimization, potential next steps could include:

Category Segmentation

- Deep-dive into mis-classified categories
- Evaluate rare category performance (considering AdeptID specializes in hard-to-fill roles and value diversities)
- For rare categories, experiment data augmentation (split body into pieces)

	Category_Name	Category_Count	Category_Accuracy
14	Anthropologists and Archeologists	2	1.0
13	Animal Trainers	2	1.0
489	Tax Preparers	6	1.0
24	Audiologists	2	1.0
485	Sustainability Specialists	2	1.0

Representation improvement

- Deep-dive into job body text

Job body may contain company description, responsibilities, qualifications, benefits apart from job description. Further NLP extraction could be helpful to improve job posting representations.

- Utilize existing entity representation spaces

If skills could be extracted from the job description, existing tools (1,000 latent skill features) could be added into feature space to test if it could improve model performance.

Model experimentation

- Context window

While the chosen model ('all-MiniLM-L6-v2') supports text up to 256 tokens, sequence length is limited to 128 tokens while training. Truncating to the recommended size (i.e. 128 tokens) gives better results than extending it to 256. (reference: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2/discussions/52)

- Other pre-trained models

Experiment other pre-trained language models (could also have different maximum sequence length, e.g., 'BAAI/bge-small-en-v1.5' has a sequence length of 512 tokens) and compare model performance.

- Multilingual situation

Experiment multilingual pre-trained model ('paraphrase-xlm-r-multilingual-v1') if necessary.

Engineering refactorization

- Embed into existing infrastructure and modules
- Save category embedding into vector store for faster inference
- Schedule model retraining for new job categories or data shift