Document Matching for Job Descriptions

Stanford CS224N Custom Project

Lum Yao Jun
Department of Computer Science
Stanford University
yjlum@stanford.edu

Abstract

We train a document encoder to match online job descriptions to one of many standardized job roles from Singapore's Skills Framework. The encoder generates semantically meaningful document encodings from textual descriptions of job roles, which are then compared using Cosine Similarity to determine matching. During training, we implement the methodology used by Sentence-BERT, fine tuning pre-trained BERT models using a siamese network architecture on labelled document pairs. Overall, we find that this method achieves better results than using off-the-shelf pre-trained BERT embeddings. Models from this paper will be used to categorize job demand in Singapore's economy for the purposes of economic surveillance and policy support.

1 Introduction and Background.

One of the key challenges in economic policy is accurately measuring the demand for various jobs. While data on hiring is readily available from both online and administrative data sources, it is generally difficult to segment job demand, due to the free text nature of job descriptions. Currently, substantial effort, through manual tagging or survey data collection, is required to accurately segment the demand for different jobs in the economy. Despite this difficulty, insight on which types of jobs are increasingly demanded would be necessary to support the crafting of education and skills training related policies.

In Singapore, the Skills Framework was created to provide a standardized reference for sectors, career pathways, job roles, and skills, with the aim of creating a common skills language for individuals, employers and training providers. The categorization of Singapore's job demand according to the Skills Framework's job roles would in turn inform key stakeholders on the types of skills more and more demanded in the economy. This would in turn guide individuals on the type of skills necessary to grade their employability, and the Singaporean government on the type of skills training programs to implement and support.

Currently, some jobs on certain job portals may be linked to the Skills Framework via a Singapore Standard Occupational Classification (SSOC) ID. However, the SSOC for an online job description is frequently unavailable as job portals generally do not require employers to specify one. Due to the large daily volume in job postings on job portals, manual tagging of SSOCs via human effort is vastly impractical. There thus exists a need to accurately match online job postings to standardized Skills Framework job roles via an algorithmic approach.

In this paper, we train a model to generate embeddings for job descriptions on job portals, for the purposes of matching these to the standardized job roles in the skills framework to support economic surveillance and policy.

2 Related Work.

There are many different training tasks in the NLP space used to evaluate the performance of models. Out of these tasks, semantic textual similarity (STS) involves determining if 2 different sentences are similar in meaning and semantics.

In 2018, the BERT model was introduced, setting records in various NLP-related tasks [1]. While BERT achieved state of the art performance in STS, the default model setup used by BERT was deemed unsuitable for finding the most similar sentence pairs, due to the need to check for all possible combinations of sentences [2].

Alternative methods used to find similar sentence pairs include comparing extracted semantically meaningful sentence embeddings from the BERT model using similarity measures [2]. In such a case, the cosine similarity between the embeddings of various descriptions of the same job should be high. However, in their paper, Reimers and Gurevych show that BERT sentence embeddings extracted in the default manner often do worse than using default GloVe embeddings [2].

Reimers and Gurevych then present Sentence-BERT (SBERT), a model to efficiently find similar sentence pairs. They construct SBERT by enhancing the original BERT model in two ways:

1. Using different network structures such as a siamese network structure, where 2 sentences are passed through pre-trained BERT models with shared/tied weights.
2. Adding a pooling operation to the output of each BERT model to derive the individual sentence embeddings.

They then show that SBERT achieves an improvement over other sentence embedding methods on various common benchmarks, while being computationally faster than comparable models. Due to these results, the methodology of this paper will aim to emulate the structure proposed by Reimers and Gurevych, applied to the matching of job descriptions.

3 Approach.

To determine the matching of job descriptions to standardized job roles, we compute the cosine similarity between the document embeddings of the job description and the textual descriptions of the standardized job roles.

To generate our document embeddings, we mirror the approach introduced by Reimers and Gurevych's paper on Sentence-BERT [2], with inputs of both matched (refer to same standardized job role) and unmatched (refer to different standardized job role) job descriptions.

To do this, we load 2 pre-trained BERT models from the Transformers library [3] in a siamese network architecture with shared weights. We then add a mean pooling layer over each BERT model to generate embeddings, with u and v indicating the embeddings for paired documents. We also experiment with two variations of model structure and objective function.

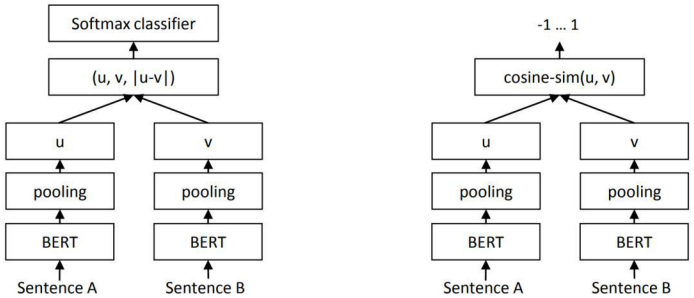

Figure 1: Model architecture used in Reimers and Gurevych's paper[2], which we follow. u and v are the extracted sentence embeddings. **LHS**:Softmax approach **RHS**:Cosine Similarity approach.

Softmax Approach. Here, the BERT model output embeddings u and v are concatenated with their element-wise difference $|u - v|$. We then feed this concatenation through a softmax function where $W_t \in \mathbb{R}^{3n*k}$:

$$o = softmax(W_t(u, v, |u - v|))$$

where n is our embeddings dimension (we use 768) and k the number of labels (k = 1 for our use case as our current outcome/true labels are binary). This variation of the model is optimized with cross-entropy loss, and is depicted on the LHS of Figure 1.

The Softmax Approach has the advantage of allowing for the possibility of multiple distinct labels for paired documents. While our current labels are binary (reflecting if paired documents reflect the same specific Skills Framework Job Role), future use cases might include determining the types of similarity between paired job descriptions. For example, additional labels might indicate if the paired job descriptions are from the same Industry/Sector, or are looking for similar skillsets. We thus examine this approach to determine the comparative efficacy for our current use case.

Cosine Similarity Approach. In this approach, instead of feeding the BERT model output embeddings through another layer, we directly compute the cosine similarity between the embeddings u and v, and then use mean square loss as the objective function. This is depicted on the RHS of Figure 1.

The advantage of the Cosine Similarity approach is twofold. Firstly, there are no additional trainable parameters apart from the pre-trained BERT models, thus requiring less time to fine-tune. Secondly, we directly optimize on the cosine similarity of our embeddings in relation to our true labels (which is how we want our embeddings to behave), and thus expect better results.

As a baseline for comparison, we obtain pre-trained, but not fine-tuned embeddings, from the Sentence Transformers library constructed by the authors of the Sentence-BERT paper [2]. The Sentence Transformers embeddings are also pre-trained for the semantic textual similarity task, and are thus likely to represent the best alternative representations publicly available online for our job description matching use case.

In the construction of our model, we obtained and loaded pre-trained BERT model parameters via the Transformers library[3], and also obtained our baseline Sentence-BERT embeddings from the Sentence Transformers library [4]. Apart from this, the data preparation, siamese network structure, fine tuning process, various model heads/objective functions and model evaluation metrics were self-coded in PyTorch.

4 Experiments.

Data. The dataset used in this paper was constructed from the following sources:

1. The Singapore Standard Occupational Classification (SSOC) ID is a standardized 5 digit identification number for classifying occupations, used for population censuses, household surveys and general administration by the Singapore government.
2. SkillsFuture is a national government-backed initiative to support the lifelong development of skills for Singaporean citizens in Singapore. As part of the SkillsFuture initiative, the Skills Framework (SF) was created to provides key information on sector, career pathways, job roles, and skills. At the data level, the Skills Framework comprises a non-exhaustive list of standardized jobs roles, alongside the textual descriptions of said job roles and the skills required for each job role. Each job role from the skills framework is officially linked to a specific SSOC ID. We use the 1466 standardized job roles in the skills framework for our reference jobs.
3. MyCareersFuture (MCF) is a job portal maintained by the Singapore government for Singapore jobs. Employers post job descriptions on the job portal detailing the job demanded, while job seekers submit their resumes to apply for these jobs. Data from this source is used as MCF job descriptions are manually tagged with SSOC IDs during time of creation for administrative purposes.

We construct the dataset using 3 months worth of MyCareersFuture job descriptions each paired with a Skills Framework job role description, with binary labels indicating a 1 if the two job descriptions share the same SSOC, and a 0 if not. This resulted in 105719 examples of paired documents, of which 10572 were used as a testing set.

Evaluation method. For the evaluation of our models, we follow the framework set by Reimers and Gurevych's Sentence-BERT paper [2] and use the the Spearman rank correlation between the both input's embedding representations cosine similarity and the true labels. This method of evaluation allows us to extend the model in future to paired sentences with ordinal, non binary labels, such as labels increasing from 0 based on the level of similarity between document pairs.

Experimental details. We ran the models with the following settings:

- Generally, models using the Cosine Similarity approach were trained for 1 epoch only while models using the Softmax approach were trained for 2 epochs. Fine tuning for higher epoch numbers did not result in any performance gain, perhaps due to the large number of examples in the training dataset.
- All models used a batch size of 4, which was the largest possible due to GPU memory limitations. Due to the size of the BERT models in the siamese network structure, larger batch sizes caused a GPU memory error. With this batch size, models took an average of 9 hours to train over a single epoch.
- We used the AdamW optimizer with a learning rate of 5e-5, similar to what was used in the Sentence-BERT paper [2]. Experiments with other optimizers and learning rates did not result in performance increases.
- Mean pooling was used to pool the final hidden state of the BERT models into our 768 length embedding vectors.
- We set the maximum wordpiece token length for the BERT models at 300 (BERT allows a maximum of 512). This number was chosen in light of model size limitations and training time taken (higher length values increase the model size and training time taken, which were constrained by available GPU memory and project timelines respectively). For comparison, the average word length of a job description in our dataset is 212 words long.

Results.

Models/Approach	Optimizer	Learning Rate	Epochs	Evaluation Metric (Test Set)
Baseline	-	-	-	0.44
Softmax	AdamW	5e-5	1	0.68
Softmax	AdamW	5e-5	2	0.76
Softmax	AdamW	5e-5	3	0.72
Cosine Similarity	AdamW	5e-5	1	0.85
Cosine Similarity	SGD	1e-4 with decay	1	0.83
Cosine Similarity	AdamW	5e-5	2	0.80

Note: Various other models with differing learning rates, model sentence lengths, batch sizes and Epochs were attempted, but did not pose any gain in evaluation results.

Examining our results, all models handily beat our baseline statistic of 0.44 Spearman rank correlation, showing that the fine tuning of our models on job description data indeed has value over default pre-trained models.

In general, we find that the Cosine Similarity approach does better overall in comparison to the Softmax Approach. This is not unexpected, given the fact that the Cosine Similarity approach directly optimizes our embeddings based on similarity, which we also evaluate upon. It is also likely that part of the optimization done in the Softmax approach involved tuning the parameters in the model's Softmax layer, which does not affect the BERT model's embedding output and therefore not tuning the embeddings as well. Training models using the Cosine Similarity approach on 2 epochs or more resulted in performance losses, due to our model overfitting on the data. Experimenting with the SGD optimizer and learning rate decay also resulted in a lower Spearman rank correlation.

Models using the Softmax approach required more epochs to train before validation loss stopped decreasing, likely due to the addition of additional trainable parameters in the Softmax layer which are not present inside the Cosine Similarity approach.

Overall, the best result we are able to achieve is a 0.85 Spearman rank correlation on the test set, using the Cosine Similarity approach trained over a single epoch. For reference, the SBERT model by Reimers and Gurevych achieve a 0.87 Spearman rank correlaton on their Semantic Textual Similarity task.

5 Additional Analysis.

Additional Accuracy Statistics. We take our best performing model and generate embeddings for the online job descriptions for the test set, and then match these to the closest standardized job role embeddings using cosine similarity. Due to the nature of the Skills Framework, certain standardized job roles are similar in nature, and thus share the same SSOC ID. We thus consider the pairing of a job description with any applicable standardized job role with the same SSOC as a match.

Overall, out of the 10572 examples in our test set, our model correctly matches 8034, amounting to 76 percent accuracy. Accuracy percentages differ across different types of standardized job roles. From our analysis, standardized job roles in specialized sectors such as Accountancy, Law and Infocomm Technology tend to do better, averaging an accuracy of 83 percent. On the other hand, standardized job roles in sectors such as Social Services, Human Resources and Transport do worse, averaging around 57 percent accuracy.

We theorize that the performance amongst these sectors differ due to differing levels of sector or job specific terms and jargon. For example, accountancy related jobs tend to reference specific terms such as accounting standards, accruals and accounts payable unique to the sector. On the other hand, human resource related jobs tend to reference less sector specific jargon, and instead mention more generic terms such as "administrative and logistical support".

Error Analysis. We randomly sample 300 job descriptions which have been predicted wrongly by our model, and discover the following:

- A large number of wrongly predicted job descriptions tend to start with company descriptions in the first paragraph of the job description, instead of elaborating on the supposed demands and requirements of the job. This causes errors as the beginning text in the job description would hold no information on the nature of the job, and thus cause noise in the model. Additionally, as our data pre-processing truncates text past the the set limit of 300 word pieces, it is possible that relevant job related information may have been truncated, instead leaving behind the company description instead. 235 of our wrongly predicted job descriptions start with company descriptions.

- A significant number of wrongly predicted job descriptions are for jobs which do not fit within any standardized job role. As the Skills Framework standardized job roles were initially created for the purpose of guiding skills training, it is thus not exhaustive across all types of jobs. In particular, the skills framework lacks certain roles that are more generic in nature, such as clerical staff or cleaning staff. Job descriptions for such roles such as Administrative Support/Clerical staff are thus matched wrongly to standardized job roles which have similar generic descriptions in their description, such as Human Resources job roles. 136 of our wrongly predicted job descriptions fall into this category.

These discoveries are inherent shortcomings in both online job description and standardized job roles data, and are difficult to account for in the current model. A possible extension to this work to account for this would be to train a separate classifier to distinguish the types of job descriptions which would be difficult to categorize. This separate model would then be able to filter out job descriptions with an abundance of generic terms, or with non job-related opening paragraphs, in order to exclude them from the job demand estimation process.

6 Conclusion.

We trained a document encoder using pre-trained BERT models on a siamese network structure with shared weights for the purpose of generating semantically relevant document embeddings, following the techniques and methods proposes in Reimers and Gurevych's Sentence BERT paper. 2 different approaches were explored, with both approaches performing significantly better than our benchmark using pre-trained SBERT embeddings. Overall, our best model does relatively well on the test set and will be proposed to be used for the actual matching of online job descriptions to standardized job roles, replacing the need for manual tagging and effort. This would in turn support economic surveillance and policy work.

Further work will be explored to extend to further analysis and use cases. For example, a future extension will involve exploring the possibility of matching textual resumes to standardized job descriptions to determine the supply of "qualified" labour for standardized job roles. Additionally, we will also explore extensions to cover our current model's shortcomings. For example, a model could be trained to identify job descriptions which do not start off with the job's requirements, but instead describe the company profile, therefore causing errors in the matching process. Having such a model would allow us to filter out job descriptions unsuitable for matching in order to reduce erroneous matches.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[2] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[3] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[4] UKPLab. Ukplab/sentence-transformers.