DataSci 266 Project Proposal

Team Members: Pascual Eley, Maria Manna, & Frank Song

Topic: Forecasting Compensation and Seniority Level through Job-Description Analysis

Research Approach: We look to explore transfer learning and task-specific fine-tuning to analyze job descriptions. Specifically, we will classify job descriptions into predicted compensation buckets, as well as based on seniority levels: analyst, associate, vice president, director, managing director, etc. The importance of this task is twofold: first, it enables hiring managers to flag job postings if the description does not align with the compensation/title. Additionally, it empowers candidates to identify jobs with misaligned descriptions, titles/ compensation allowing them to target good opportunities, or avoid bad ones. One of the challenges lies in the high job description variability across different industries and companies, and conversely in the similar language used to describe desired hard and soft skills (e.g., 'driven', 'personable', 'experienced', etc).[1] [2]

Data: We plan to use the "US Job Postings from 2023-05-05" dataset, a pre-compiled and labeled job posting dataset found on Kaggle.[3] It contains 33,064 US job postings scraped from 29 sources. By leveraging transfer learning, we will reduce the need for a large corpus of labeled data.[4] [5] We will utilize the job posting body as text input, and labeled salary/job level/industry as a ground truth label to compare against.

Technique: We are currently exploring the implementation of BERT or GPT-3 for our model.[6] [7] Furthermore, we are considering using neural networks for our task, specifically CNNs or RNNs.[8] As part of the model-building process, we need to determine the salary bucket size we will use (to convert compensation into a categorical label) and the level of detail for the job categories (in order to balance the granularity of label categories with the usefulness of the categorization). Additionally, we need to determine how to weigh the accuracy of multiple outputs when measuring the overall accuracy of our model. We may find that it's more important, or feasible, to train a model that consistently categorizes job levels correctly and struggles with salary range buckets or vice versa. In exploring this, we will evaluate why the selected model architecture performs the way it does with respect to these concerns, and adjust the architecture accordingly.

---

[1] SkillSpan: Hard and Soft Skill Extraction from English Job Postings - ACL Anthology,
[2] Fine-Grained Extraction and Classification of Skill Requirements in German-Speaking Job Ads
[3] US Job Postings from 2023-05-05
[4] Transfer learning for multilingual vacancy text generation
[5] Kompetencer: Fine-grained Skill Classification in Danish Job Postings via Distant Supervision and Transfer Learning - ACL Anthology
[6] Competence-Level Prediction and Resume & Job Description Matching Using Context-Aware Transformer Models
[7] Automated Resume Screening Using Natural Language Processing
[8] Competence-Level Prediction and Resume & Job Description Matching Using Context-Aware Transformer Models

Additional Options to Extend Difficulty/Scope:

1. **Advanced analysis of model behavior:** We will spend time analyzing specific aspects of the model performance. We might identify certain areas where the model does not perform well by deep diving into prediction accuracy distribution, and then test hypotheses by tailoring specific input job descriptions that we expect to result in a particular prediction.
2. **Text Generation:** Going beyond the multiple classification task, which we foresee could be accomplished using an encoder-only architecture with a FNN on top, we would also explore the possibility of a follow-on text generation task. This task would generate a potential job description based on the input salary/job level/industry/function. We could use the same dataset from the classification task, inverting the inputs and labels to compare generated text against ground truth descriptions. We might first train our classification model, then use job descriptions that generated the highest accuracy and confidence in that model to train a decoder language model for text generation. The motivating idea behind this would be to create a multi-stage NLP model that predicts job characteristics (salary/level/industry/function) based on description, then compares against an input label and generates a new suggested description if there is a mismatch between predicted and actual job characteristics.