15.773 - Hands-on Deep Learning Final Project Report

Detecting Fake Job Postings with Deep Learning

Radhika Anbazhagan, Clemence Couteau, Arya Ondaatje, Reid Silverhart

## Problem Description

*Defining the Issue of Fake Job Postings*

Fake job postings can generally be categorized into two types: scammer job postings and company-generated fake job postings (often called "ghost jobs"). Scammer job postings are fraudulent listings created with the intent to deceive job seekers, often promising unrealistically high salaries or requiring upfront payments for job placement. These scams frequently aim to steal personal information or financial details.

On the other hand, “ghost jobs” are job postings created by companies with no intention of actually hiring. These may be used to create the illusion of company growth, collect resumes for a future talent pool, fulfill hiring quotas, even as a tactic to make current employees feel replaceable.[[1]](#footnote-1) The prevalence of fake job postings has surged significantly, with studies indicating that up to 40% of companies have posted fake job listings in the past year.[[2]](#footnote-2) While not always malicious, ghost jobs can mislead applicants and create frustration in the job market.

*Implications for Job Seekers and Employers*

For job seekers, fake job postings – whether scams or ghost jobs – can be incredibly frustrating, disheartening, and even dangerous. Job seekers invest significant time and effort in crafting applications, preparing for interviews, and following up on job listings that turn out to be fake. Repeated encounters with fake job postings can erode trust in the job market and in specific companies, making job seekers skeptical of legitimate opportunities. Scam job postings can lead to financial loss or even identity theft.

For employers, maintaining transparency in hiring is essential for company reputation. While posting fake jobs may create the perception of growth and stability initially, they can severely damage a company’s reputation if discovered. The presence of ghost jobs can create skepticism among candidates, while scam postings on a company's job board can damage credibility if not addressed.

*Existing Challenges*

Identifying and addressing fake job postings present several challenges. Fake job postings often mimic legitimate ones, making them difficult to distinguish. They may use professional language and even company logos. Many job boards and platforms lack stringent verification processes, allowing fake listings to slip through. Scammers continuously adapt their techniques, making it essential for detection systems to evolve accordingly.

*Importance of Automated Detection*

Automated detection of fake job postings is crucial for several reasons. It can quickly analyze large volumes of job postings, identifying potential fakes more efficiently than manual methods. Models can detect subtle patterns and linguistic cues that indicate fraudulent postings, improving the accuracy of detection. Automated detection helps protect job seekers from falling victim to scams, safeguarding their personal information and financial security. Implementing robust detection mechanisms can help restore trust in online job platforms and the job market as a whole.

*Project Objectives*

The primary objective of this project is to analyze both types of fake job postings – scammer ads and company-generated ghost jobs – using Kaggle datasets to identify patterns and distinguishing characteristics. The project aims to develop an automated detection system capable of accurately classifying job ads as genuine or fraudulent. Additional objectives include benchmarking the performance of traditional machine learning methods against deep learning models (e.g., BERT), exploring challenges related to class imbalance and generalization to unseen data, and reflecting on the trade-offs between model complexity and deployment feasibility. By achieving these objectives, the project aims to contribute to a more transparent and trustworthy job market, benefiting both job seekers and employers.

## Approach

*Data Description*

Our project utilizes two datasets from Kaggle:

1. [**The Fake Job Postings Dataset**](https://www.kaggle.com/datasets/srisaisuhassanisetty/fake-job-postings/data) – This dataset contains 10,000 fraudulent job postings, which exhibit characteristics typical of scam job ads. These postings frequently offer high salaries with minimal qualifications, use vague job descriptions, or include requests for upfront payments.
2. [**The Real and Fake Job Postings Dataset**](https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction/data) – This dataset includes 17,880 job postings, of which about 800 (4.5%) are fake and the remaining are real. We observed that the fake listings in this dataset appeared more representative of company-generated ghost jobs rather than scam postings. These postings tend to be professionally worded and mirror legitimate job ads but are not intended to be filled.

Each dataset includes multiple fields describing job postings, such as: Title, Description, Requirements, Company Profile, Location, Employment Type, and Industry. Both datasets include a "fraudulent" column, which serves as the target variable. This binary label denotes whether a job posting is real (0) or fake (1).

*Data Preprocessing*

To prepare the datasets for training, we first merged them into a single dataset to ensure that our deep learning model was trained on a broader spectrum of fake job postings. This approach allows us to develop a classifier capable of distinguishing *both* scam job ads and company-generated ghost jobs, increasing its overall effectiveness in real-world applications. After merging, we selected key fields – title, description, requirements, and fraudulent – while removing entries with missing values in these essential columns. To prevent data leakage and improve model generalization, duplicate job postings were identified and removed.

Next, we combined text fields to create a unified representation of each job posting. Specifically, we concatenated the title, description, and requirements columns into a single text string per entry. This transformation allowed the model to capture the full context of a job listing in a single input sequence, ensuring that important details across different sections were not lost.

For text processing, we applied two separate approaches depending on the model type. For the baseline regression model, we performed traditional NLP processing by removing stop words and used Bag-of-Words vectorization with a vocabulary capped at 5,000 features to represent job descriptions as word-count-based vectors. For the BERT model, we applied BERT tokenization with WordPiece encoding, which segments text into subword units while preserving contextual structure.

Finally, we split the processed dataset into 80% training, 10% validation, and 10% test sets, ensuring a balanced class distribution across all subsets.

*Baseline Model*

As a baseline, we implemented a logistic regression model using a Bag-of-Words (BoW) representation of the combined text fields (title, description, and requirements). The text was first vectorized using CountVectorizer, which converts words into frequency-based numerical representations. Stop words were removed, and the vocabulary size was limited to 5,000 features to reduce dimensionality.

Once vectorized, the transformed text was used to train a logistic regression classifier with balanced class weights to mitigate the effects of class imbalance. The model was optimized using L2 regularization and trained with a maximum of 1,000 iterations to ensure convergence. Performance was evaluated using standard classification metrics, including accuracy, precision, recall, F1-score, and ROC-AUC.

*Deep Learning Model*

To capture deeper contextual and semantic information, we fine-tuned a BERT-based model for binary classification. We used the BERT-base-uncased model from Hugging Face’s transformers library, which was pre-trained on a large corpus of English text. The title, description, and requirements fields were concatenated into a single input string per job posting and tokenized using BERT’s WordPiece tokenizer.

The tokenized input was fed into the BERTForSequenceClassification model, which added a classification head on top of the pre-trained transformer layers. The model was trained using binary cross-entropy loss and optimized with AdamW, a variation of Adam with weight decay. We trained the model for three epochs with an 80/10/10 train-validation-test split, ensuring class balance across all subsets.

Fine-tuning was performed using the Hugging Face Trainer API, leveraging gradient accumulation and automatic mixed precision (AMP) for computational efficiency. Model performance was evaluated on the validation set using accuracy, precision, recall, F1-score, and ROC-AUC, with a final assessment on the test set to measure generalization performance.

Results

### *Baseline Model: Logistic Regression with Bag-of-Words*

As a baseline, we implemented a logistic regression model using a Bag-of-Words (BoW) representation of the combined text fields (title, description, and requirements). Originally, due to an issue combining the two datasets, our dataset only had ~5% fake job listings, and the model performed poorly due to the imbalance. After solving this issue and combining the two datasets, the baseline model achieved strong results. The model demonstrated an excellent ability to distinguish between fake and real job postings based on simple word frequency patterns.

Baseline Performance (Logistic Regression)

|  |  |
| --- | --- |
| **Metric** | **Value** |
| **Accuracy** | 0.9812 |
| **Precision** | 0.9863 |
| **Recall** | 0.9678 |
| **F1-Score** | 0.9770 |
| **ROC-AUC** | 0.9924 |

### Confusion Matrix (Logistic Regression)

|  |  |  |
| --- | --- | --- |
|  | **Predicted Real** | **Predicted Fake** |
| **Actual Real** | 2419 | 23 |
| **Actual Fake** | 55 | 1655 |

While Bag-of-Words is a relatively simple feature extraction method, logistic regression achieved high precision and recall, likely benefiting from the dataset's clearer class separation after addressing class imbalance. However, it still incurred a significant number of false negatives (55), indicating that it does not do a perfect job identifying fake postings.

### *Deep Learning Model: Fine-Tuned BERT Transformer*

To capture deeper contextual and semantic information, we fine-tuned a pre-trained BERT-base-uncased model on the same balanced dataset. This model was also trained exclusively on text-based features, including the title, description, and requirements fields, without using metadata (e.g., "has\_company\_logo," "employment\_type").

The BERT model was trained over 3 epochs with an 80/10/10 train/validation/test split, using stratified sampling to maintain class balance across splits.

The BERT model slightly outperformed the logistic regression model across all key metrics. By leveraging contextual embeddings, BERT captured deeper semantic relationships within job descriptions.

Deep Learning Model Performance (Fine-Tuned BERT)

|  |  |
| --- | --- |
| **Metric** | **Value** |
| **Accuracy** | 0.9834 |
| **Precision** | 0.9887 |
| **Recall** | 0.9708 |
| **F1-Score** | 0.9796 |
| **ROC-AUC** | 0.9967 |

### Confusion Matrix (Fine-Tuned BERT)

|  |  |  |
| --- | --- | --- |
|  | **Predicted Real** | **Predicted Fake** |
| **Actual Real** | 2423 | 19 |
| **Actual Fake** | 50 | 1660 |

The BERT model reduced both false positives (19) and false negatives (50) relative to logistic regression, leading to a slight but meaningful improvement in both F1-score and ROC-AUC. This improvement suggests that contextual embeddings captured by BERT were more effective at distinguishing subtle differences between real and fake job postings.

*Comparison Summary*

|  |  |  |
| --- | --- | --- |
| **Metric** | **Logistic Regression** | **BERT Transformer** |
| **Accuracy** | 0.9812 | 0.9834 |
| **Precision (fake)** | 0.9863 | 0.9887 |
| **Recall (fake)** | 0.9678 | 0.9708 |
| **F1-Score (fake)** | 0.9770 | 0.9796 |
| **ROC-AUC** | 0.9924 | 0.9967 |

### *Takeaway*

Balancing the dataset significantly improved the baseline model’s performance, allowing it to distinguish between real and fake job postings more effectively. However, BERT still maintained a clear advantage by capturing more nuanced linguistic and contextual cues, resulting in fewer misclassifications overall. The deep learning model reduced both false positives and false negatives, making it more reliable for detecting fraudulent job postings. While the numerical difference in performance may seem modest, in high-stakes applications—where even a handful of missed fraudulent postings can erode user trust and platform integrity—BERT’s improvements are meaningful and impactful. That said, the 50 false negatives highlight an opportunity for further refinement, such as additional training data, hyperparameter tuning, or alternative deep learning architectures.

## Lessons Learned

### *1. Balancing the Dataset Made a Major Impact*

One of the most important insights from this project was how class imbalance influenced model performance. Initially, due to the imbalance in the *Real and Fake Jobs* dataset, there were very few fake postings, leading to models that struggled with recall on fraudulent jobs. By correctly integrating the additional dataset containing 10,000 fake postings, we ensured a more even class distribution. This adjustment significantly improved both the logistic regression and BERT models, reinforcing how critical balanced and representative data is in fraud detection tasks. It also underscores the need for careful dataset curation when working with real-world classification problems.

### *2. Strong Baseline Performance but Limitations in Recall*

The logistic regression baseline performed surprisingly well after balancing the dataset, achieving high accuracy (98%) and a strong F1-score (97.7%). This suggests that in some cases, simpler models with Bag-of-Words representations can be effective, particularly when classes are well-separated. However, despite its strong precision, logistic regression still suffered from higher false negatives than the BERT model. This is likely due to its reliance on word frequency patterns rather than contextual meaning, making it less effective in detecting subtly deceptive job postings.

### *3. BERT Provides Incremental but Meaningful Gains*

While the performance gap between the baseline and BERT model narrowed after balancing, BERT still showed marginally superior precision, recall, and ROC-AUC, reducing both false positives and false negatives. The contextual embeddings provided by BERT proved valuable for capturing subtleties in the text, resulting in a more robust model, particularly useful for edge cases and ambiguous postings.

### *4. Practical Implications*

The project highlighted the trade-offs between model complexity and real-world deployment. Logistic regression offers simplicity and faster inference times, while BERT's enhanced accuracy may justify the added computational cost in environments where fraudulent job detection carries higher risk or reputational impact.

## Next Steps

### *1. Further Hyperparameter Tuning for BERT*

Future iterations should include a more thorough hyperparameter search for the BERT model, exploring variations in learning rates, batch sizes, and fine-tuning epochs to optimize performance, particularly for improving recall on fake job postings.

### *2. Exploring Lightweight Models*

Given the trade-off between accuracy and computational cost, experimenting with lighter models such as DistilBERT or ALBERT could provide faster inference times while maintaining competitive performance.

*3. Robustness Testing on External Dataset*

An important next step is to test the model on fully out-of-sample data, such as an external fake job dataset or live job board postings, to evaluate its generalization ability beyond the combined dataset used in this project.

### *4. Revisiting Retrieval-Augmented Techniques*

While we initially proposed implementing a Retrieval-Augmented Generation (RAG) approach, the project pivoted toward refining the BERT classifier pipeline. In the future, incorporating a retrieval + augmentation strategy (e.g., retrieving similar postings using Sentence-BERT) could enhance the model’s ability to leverage external context and improve predictive accuracy, especially for borderline cases.

## Google Colab Link

<https://colab.research.google.com/drive/1QDN10sVM5MshlVgXdNw_do3mPfhyx82_#scrollTo=kAz4qY3WliYt>

1. https://www.forbes.com/sites/rachelwells/2024/08/13/36-of-job-adverts-are-fake-how-to-spot-them-in-2024/ [↑](#footnote-ref-1)
2. https://www.cbsnews.com/news/fake-job-listing-ghost-jobs-cbs-news-explains/ [↑](#footnote-ref-2)