Evaluating Resume Efficacy and Optimal Features Using LLM

Caleb Wiebolt
wiebo034@umn.edu

Ross Volkov
volko032@umn.edu

Ben Davidson
david968@umn.edu

Abstract

In our project *Evaluating Resume Efficacy and Optimal Features Using LLM* we sought to understand how Large Language Models screened resumes. We fine-tuned two resume classifiers with RoBERTa and Longformer LLMs and evaluated their inferences with Shapely values. The models achieved a classification accuracy of 0.92 across 24 career classes. The models were great at predicting distinct classes, but struggled with those which were closely related to other classes. Our tests with malicious prompting indicated that models could be swayed by injecting buzzwords into a resume, and the models had a significant bias toward tokens at the beginning of a resume. Finally, we explored the ethical implications of AI resume screening, and compared our model's reasoning to human reasoning.

1 Introduction

The integration of Natural Language Processing and machine learning technologies in recruitment processes is a growing trend in the job application process. Many companies large and small are already using complex resume screening pipelines that involve classifying and ranking job applicants by their resumes. Various research studies have focused on this intersection, each applying unique methodologies and focusing on different aspects of the recruitment process. These models are in widespread use but are they capable of replacing human applicant screening? Are they making decisions that are moral and just? Our project, "Evaluating Resume Efficacy and Optimal Features Using LLM," focuses on understanding the internal decision-making processes of automated resume screening models. It aims to train a model to identify the targeted job of a resume and uses visualization techniques to analyze how the model makes its decisions, allowing for an exploration of biases and ethical considerations in the model's outputs.

We seek to explore how these models work, how accurate they are in predicting the intent of a resume, and can adversarial resumes trick the models. For the companies that use these systems this type of exploration is vital to ensure a lack of bias in the hiring process which might ignore qualified candidates or at worse cause legal liability for discriminatory hiring. For the applicants themselves, understanding these models can open the doors to writing resumes that are better tailored to present the applicant in the best light possible.

2 Literature Survey

There is a wealth of published literature on the topic of using NLP in resume screening. Before we started our experiments we wanted to get an understanding of the current state of the field by surveying various papers. A study detailed in one paper focused on the extraction and categorization of skills from resumes and job descriptions using NLP, laying the groundwork for more advanced matching algorithms (Gopalakrishna and Varadharajan, 2019). The authors of this paper utilized a rule-based extraction method to identify skills from the textual data and employed clustering techniques to categorize these skills (Gopalakrishna and Varadharajan, 2019). Their findings provided a scalable way to handle large volumes of resumes, and the results indicated a significant improvement in the precision of skill categorization compared to previous models. In their analysis, however, they did not go beyond simply matching skills and job descriptions. Another research paper we read proposed a system that uses NLP to measure the similarity between job posts and resumes, aiming to make the recruitment process more efficient (Sultana et al., 2023). In this paper, the researchers implemented various NLP models, including the GloVe model, and applied similarity measures such as cosine similarity and euclidean distance (Sultana et al., 2023).

Their findings showed that the GloVe model outperformed others with a best accuracy of 79.2 percent, suggesting its effectiveness in matching jobs with resumes (Sultana et al., 2023). However, this approach primarily focused on the practical aspects of matching, without deeply looking into what exactly in the text caused the model to give specific classifications.

In another paper, Grech and Suda presented a comprehensive framework for evaluating various matching algorithms, focusing on their performance and accuracy (Grech and Suda, 2020). They used a multi-criteria decision-making approach to assess different algorithms and introduced an evaluation metric that could better reflect the performance of the matching algorithms in a practical setting (Grech and Suda, 2020). The results highlighted the challenges in achieving high accuracy in the matching process and the necessity for more robust and complex models. Another paper emphasizes the importance of explainability and traceability in automated recruitment systems (Barrak, 2021). This research explored how modern language models could be combined with ontologies and knowledge bases, aiming to create an automated recruitment system that provides detailed matching explanations (Barrak, 2021). They planned to evaluate their system's performance using a gold dataset and the metric of normalized discounted cumulative gain (Barrak, 2021). The authors focus on making the model's decision-making process more transparent and understandable is both timely and similar to our own stated goals.

Given the literature surveyed our project fits nicely with existing research in its application of NLP in recruitment processes and its goal to improve these processes. However, our project distinguishes itself by focusing on exploring and understanding the decision-making processes of automated resume-screening models. While there are clearly many avenues of study in this field, we plan to focus our efforts on the exploration and analysis of model decision-making, biases, and the ethical considerations that can come from the use of these types of models in the context of resume screening.

3 Our Methods

Our goal was to recreate a part of a resume processing pipeline, specifically a system that classifies resumes into job categories. To do this we fine-tuned Large Language Models (LLM) to determine which roles an applicant was best suited for. Then, we analyzed what tokens caused the model to give the classification it did. Our initial hypothesis was that job experience and education would be the biggest indicators of what job a resume is used to apply for. We also theorized that the model would be easily fooled by "buzzwords" related to the specific job categories. To facilitate our plan we used a dataset of template resumes.

3.1 Parsing and Data Cleanup

The resume dataset citation has a collection of 2400 resumes each belonging to one of 24 unique classes representing the job which the resume applied for figure. Each resume was given in PDF form, html form and raw text.

Category	Resume Count
INFORMATION-TECHNOLOGY	120
BUSINESS-DEVELOPMENT	120
FINANCE	118
ADVOCATE	118
ACCOUNTANT	118
ENGINEERING	118
CHEF	118
AVIATION	117
FITNESS	117
SALES	116
BANKING	115
HEALTHCARE	115
CONSULTANT	115
CONSTRUCTION	112
PUBLIC-RELATIONS	111
HR	110
DESIGNER	107
ARTS	103
TEACHER	102
APPAREL	97
DIGITAL-MEDIA	96
AGRICULTURE	63
AUTOMOBILE	36
BPO	22

Table 1

To preprocess the data, we removed all non-English symbols, whitespace, URL's, tags, and mentions. We then tokenized the data in sentences and words to pass into the model. The dataset was shuffled and split 90-10 into training and validation sets.

3.2 Fine Tuning Our Model

Initially, we used RoBERTa (Liu et al., 2019) as the base of our fine-tuned model, however with its token limit we were forced to truncate large portions of our dataset. To ease this issue we later switched to Longformer (Beltagy et al., 2020) as our base model, utilizing its much larger token limit to process much more of each resume.

We trained the RoBERTa and Longformer models on the training set for 3 epochs each. Training was terminated when the models began to overfit on the data. The inputs to the RoBERTa model were capped at 500 tokens, while the longformer model had inputs capped at 4096 tokens.

To determine the optimal tokens for each class, we used shapely values with the SHAP Python library. Shapely values determine the effect each individual input has on the overall prediction using a permutation-based approach. For a given resume, SHAP will withhold a token in the resume, and make predictions with various combinations of other tokens. By comparing the output scores, the effect of that input on each class prediction can be calculated.

We ran SHAP evaluation for each resume in the test dataset and saved their SHAP outputs as interactive HTML pages.

4 Results

We used a formula: accuracy $= \frac{\text{Number correct}}{\text{Total predictions}}$ to determine the accuracy of our models (Table 2). The Longformer model outperformed the RoBERTa model by 2%.

the average resume length was 1056 tokens, meaning that only roughly half of the average resume was able to be processed by RoBERTa. Longformer has a max input length of 4096 tokens, so 99% of input resumes were able to be processed (Table 2).

4.1 Evaluating SHAP outputs

We utilized Shapely values to investigate successful and unsuccessful predictions. In the figures, the red highlight is a positive indication of the selected class, and the blue is a negative indication. The relative intensity of the correlation is shown by the intensity of the highlight.

The distribution of classes affected the output scores and the accuracy of the model. For certain classes, such as INFORMATION-TECHNOLOGY, CHEF, and AVIATION, the model had very high average confidence scores, often between 0.85 and 0.75. Alternatively, classes such as HR, SALES, BUSINESS-DEVELOPMENT, and CONSULTING had much lower average confidence scores, often between 0.20 and 0.30. In these instances, there was less certainty of the classification between the classes. We believe that this is due to the fact that many of those classes are closely related in job type and relevant skills. Thus, resumes for those jobs are very similar.

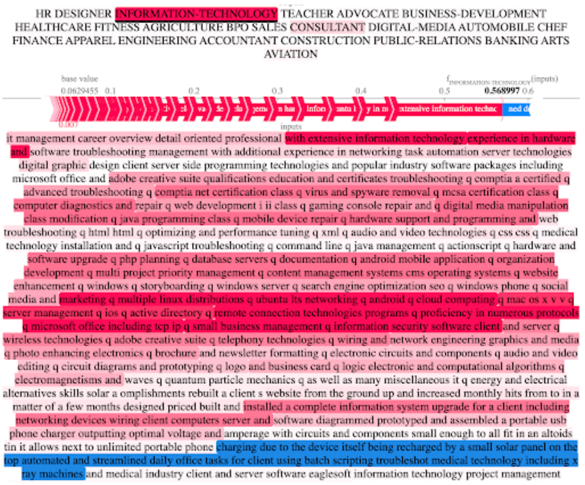
An example of a correct prediction of the INFORMATION-TECHNOLOGY class is shown in Figure 1. Sentences with software-related language such as "ubuntu", "networking", "remote" were identified to be positively correlated with the INFORMATION-TECHNOLOGY class. This is largely expected behavior, for humans likewise associate those words with INFORMATION-TECHNOLOGY. Contrastingly, the model was incorrect in its analysis of the final sentence. The sentence is highlighted intensely blue, but it mentions batch-scripting and troubleshooting medical technology. The "medical technology" tokens likely confused the model and caused it to believe that this part was closer related to the healthcare field. This confusion would likely be replicated by humans because it is not clear from that sentence if the person is working a role more focused on healthcare, or programming.

An example of an unsuccessful prediction of the CONSULTANT class is shown in figure 2. The model predicted the resume class to be BUSINESS-DEVELOPMENT with a confidence score of 0.54. The confidence score for CONSULTANT was just 0.024. In the figure, it is shown that the first line of their resume, which states "business consultant professional summary" was shown to be a negative indicator of the CONSULTANT class. Meanwhile, it was the strongest predictor of the BUSINESS-DEVELOPMENT class. We believe that this is similar to the confusion that the model had in the previous example, where the "medical technology" phrase skewed the predicted class away from IT, despite the words "scripting" and "automated" also being present (figure 1). This indicates a clear discrepancy between human and machine reasoning. Humans would immediately see the headline "consultant" and classify the resume as such, but machines can be thrown off by tokens that indicate similar classes like "business".

Shapely values were also used to calculate the greatest token predictors for each class (figures 3,

	Model Accuracy	Max input length	Percent of resumes that were truncated
RoBERTa	0.90	512	98.9%
Longformer	0.92	4096	49.7%

Table 2: The table shows the model accuracies in relation to the input length



HR DESIGNER INFORMATION-TECHNOLOGY TEACHER ADVOCATE BUSINESS-DEVELOPMENT HEALTHCARE FITNESS AGRICULTURE BPO SALES CONSULTANT DIGITAL-MEDIA AUTOMOBILE CHEF FINANCE APPAREL ENGINEERING ACCOUNTANT CONSTRUCTION PUBLIC-RELATIONS BANKING ARTS AVIATION

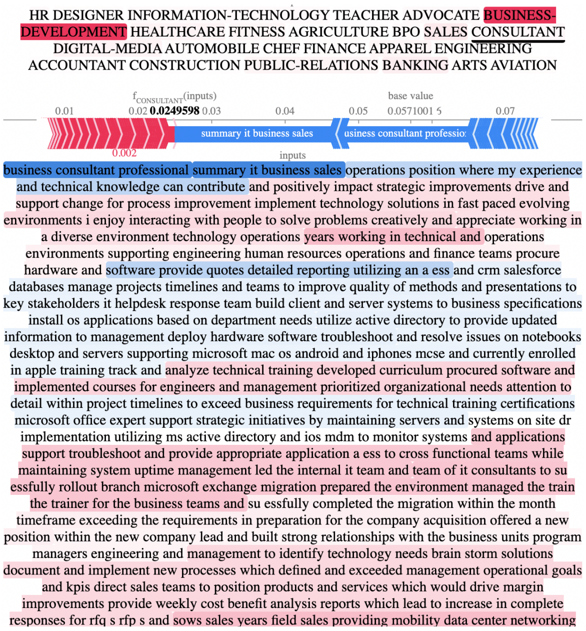
it management career overview detail oriented professional with extensive information technology experience in hardware and software troubleshooting management with additional experience in networking task automation server technologies digital graphic design client server side programming technologies and popular industry software packages including microsoft office and adobe creative suite qualifications education and certificates troubleshooting q comptia a certified q advanced troubleshooting q comptia net certification class q virus and spyware removal q mcsa certification class q computer diagnostics and repair q web development i ii class q gaming console repair and q digital media manipulation class modification q java programming class q mobile device repair q hardware support and programming and web troubleshooting q html html q optimizing and performance tuning q xml q audio and video technologies q css css q medical technology installation and q javascript troubleshooting q command line q java management q actionscript q hardware and software upgrade q php planning q database servers q documentation q android mobile application q organization development q multi project priority management q content management systems cms operating systems q website enhancement q windows q storyboarding q windows server q search engine optimization seo q windows phone q social media and marketing q multiple linux distributions q ubuntu lts networking q android q cloud computing q mac os x v v q server management q ios q active directory q remote connection technologies programs q proficiency in numerous protocols q microsoft office including tcp ip q small business management q information security software client and server q wireless technologies q adobe creative suite q telephony technologies q wiring and network engineering graphics and media q photo enhancing electronics q brochure and newsletter formatting q electronic circuits and components q audio and video editing q circuit diagrams and prototyping q logo and business card q logic electronic and computational algorithms q electromagnetisms and waves q quantum particle mechanics q as well as many miscellaneous it q energy and electrical alternatives skills solar a omplishments rebuilt a client s website from the ground up and increased monthly hits from to in a matter of a few months designed priced built and installed a complete information system upgrade for a client including networking devices wiring client computers server and software diagrammed prototyped and assembled a portable usb phone charger outputting optimal voltage and amperage with circuits and components small enough to all fit in an altoids tin it allows next to unlimited portable phone charging due to the device itself being recharged by a small solar panel on the top automated and streamlined daily office tasks for client using batch scripting troubleshot medical technology including x ray machines and medical industry client and server software eaglesoft information technology project management

Figure 1: The above graph shows the token heatmap of a successful prediction of the IT class

4). Many of the best tokens were reasonable indicators of the class, but there were many noisy tokens as well.

An example of reasonable best token predictions are shown for the DESIGNER class (figure 4), Tokens like "designer", "jewelry", "cad" and "graphic" are all tokens that humans would also associate with the class. The other tokens like "surviv", "alot" and "exp" would not be words that humans would typically associate with DESIGNER.

An example of noisy best token predictors is shown for the FINANCE class (figure 3). There were only 3 singular tokens overall that had positive correlations with the FINANCE class, and of those 3, "remarkably" had the strongest positive correlation. Ironically, "finance" itself had less of a positive correlation. This is likely due to the presence of "remarkably" in a few of the finance resumes. Since this token was disproportionately present in those resumes, the model weights it very high. We think that there is great promise in using these tools to help students and young professionals evaluate their own resumes. While not a full-featured tool, these visualizations are incredibly useful and can show which tokens one should consider emphasizing for a certain job.



HR DESIGNER INFORMATION-TECHNOLOGY TEACHER ADVOCATE BUSINESS-DEVELOPMENT HEALTHCARE FITNESS AGRICULTURE BPO SALES CONSULTANT DIGITAL-MEDIA AUTOMOBILE CHEF FINANCE APPAREL ENGINEERING ACCOUNTANT CONSTRUCTION PUBLIC-RELATIONS BANKING ARTS AVIATION

business consultant professional summary it business sales operations position where my experience and technical knowledge can contribute and positively impact strategic improvements drive and support change for process improvement implement technology solutions in fast paced evolving environments i enjoy interacting with people to solve problems creatively and appreciate working in a diverse environment technology operations years working in technical and operations environments supporting engineering human resources operations and finance teams procure hardware and software provide quotes detailed reporting utilizing an a ess and crm salesforce databases manage projects timelines and teams to improve quality of methods and presentations to key stakeholders it helpdesk response team build client and server systems to business specifications install os applications based on department needs utilize active directory to provide updated information to management deploy hardware software troubleshoot and resolve issues on notebooks desktop and servers supporting microsoft mac os android and iphones mcse and currently enrolled in apple training track and analyze technical training developed curriculum procured software and implemented courses for engineers and management prioritized organizational needs attention to detail within project timelines to exceed business requirements for technical training certifications microsoft office expert support strategic initiatives by maintaining servers and systems on site dr implementation utilizing ms active directory and ios mdm to monitor systems and applications support troubleshoot and provide appropriate application a ess to cross functional teams while maintaining system uptime management led the internal it team and team of it consultants to su essfully rollout branch microsoft exchange migration prepared the environment managed the train the trainer for the business teams and su essfully completed the migration within the month timeframe exceeding the requirements in preparation for the company acquisition offered a new position within the new company lead and built strong relationships with the business units program managers engineering and management to identify technology needs brain storm solutions document and implement new processes which defined and exceeded management operational goals and kpis direct sales teams to position products and services which would drive margin improvements provide weekly cost benefit analysis reports which lead to increase in complete responses for rfq s rfp s and sows sales years field sales providing mobility data center networking

Figure 2: The model incorrectly predicted CONSULTANT, ground truth was BUSINESS-DEVELOPMENT

4.2 Malicious Prompting and Prompt Injection

In the interest of further understanding our model and its possible shortcomings and their impacts. We decided to test the effect that Malicious Prompting, such as prompt injection had on the classification output. In order to do this we chose a class we had few examples of in our evaluation set, HR. Then we created two prompts to be appended on to the end of each resume in the evaluation set that was not targeting an HR position. The first prompt was based on instruction injection and was a simple sentence instructing the model to ignore all the previous input and to classify the resume as being for an HR job. The second prompt was created by looking at resumes targeting HR positions, running them through our classifier, and aggregating the most weighted sentences in those resumes. In this way, we ended up with a 604 word string of sentences that the model considered to be strong indicators of the resume belonging to the HR class. We then set up a script to run through the entire evaluation set, acquiring a classification score for the HR class with no injection appended, the in-

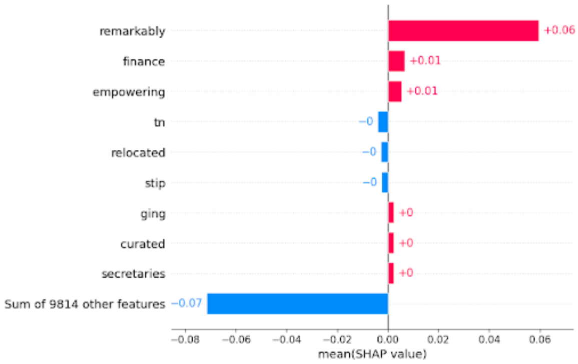


Figure 3: The above graph shows best token predictors of the finance class

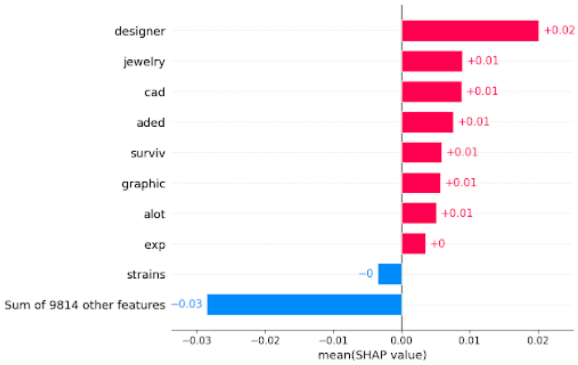


Figure 4: The above graph shows the best token predictors of the designer class.

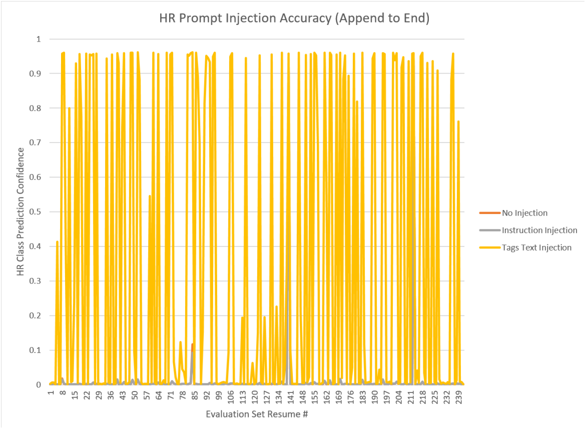


Figure 5: The above graph shows the results of our prompt injection experiment run over the evaluation set of resumes. The prompts were appended to the end of the preexisting resume text.

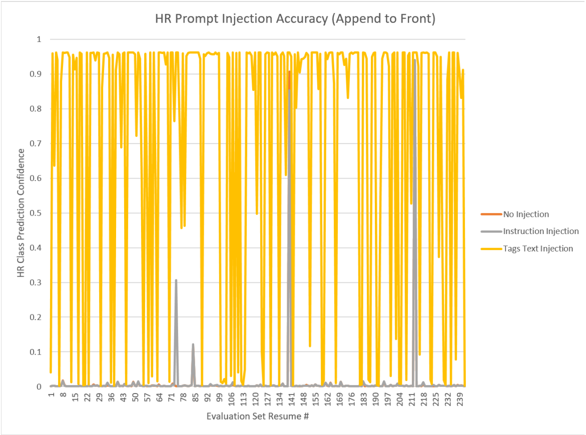


Figure 6: The above graph shows the results of our prompt injection experiment run over the evaluation set of resumes. The prompts were appended to the front of the preexisting resume text.

struction prompt appended, and the tags prompt appended. Figure 5, shows the graphed output of that experiment.

From the graph we can see that the instruction injection prompt had little to no influence on the classification score of each resume, with one or two outlying exceptions. As for the Tag Text Prompt, it seemed to have a more prominent effect but only on some resumes. The data also seems to indicate an all-or-nothing effect, with the classification score either being very slightly affected or the classification shifting to an extreme amount. One cause of this is likely the strength of the resume's original classification. Resumes with weaker initial classification scores are much easier to sway with the prompt injection.

Beyond the strength of the injected text itself, however, we were curious what role order was playing in the efficacy of our injection methods. To investigate this more fully we repeated our experiment, this time appending the prompts to the beginning of the resume text. Figure 6 shows the results of this experiment.

Based on Figure 2 we see that the order in which the text is evaluated by our model has a large effect on the the classification score that the model gives. By appending our prompts to the front of the text, we see a mild improvement in the performance of our instruction prompt and a large improvement when using our Tag Text prompt. This implies that our model tends to bias towards tokens appearing near the beginning of the text. This finding is further reinforced when we consider the average differences between the baseline no-injection scores and the injection counterpart scores for both experiments, shown in Table 3.

From these experiments, it is clear that prompt injection methods can be quite effective in convincing our model to misclassify text. This is only true however if the malicious prompt takes the form of a sizeable amount of text targeted at a specific

	Avg. Delta w/Instructions Prompt	Avg. Delta w/ Tags Text Prompt
Appended to Front	0.001871306	0.703657304
Appended to End	0.000310083	0.330112325

Table 3: The table shows the average change in classification score using each prompt and appending that prompt to either the front of the resume or the end.

classification. When smaller prompts written in an instruction style are used the effect on the overall classification is minimal. This is most likely due to our model not being fine-tuned for instruction following. While this is a positive development, as larger chunks of text designed to sway a model might be harder to hide within a document using methods like small white font, it does further reveal the reality that our model does not seem to understand the document in a holistic sense.

These experiments also reveal a strong bias in our model towards tokens near the beginning of the text being evaluated. This kind of first come first serve bias is incredibly problematic given the varying nature of the layout of different resumes. A resume with its experience section at the top of the page would be considered significantly differently than the same resume with that experience section positioned at the bottom. This is further evidence of our model's inability to actually comprehend the document as a whole as a human would.

4.3 Comparison with GPT-4

We also compared the resume classification similarities and differences of GPT-4 with the LLMs used in our project. Both GPT-4 and the LLMs demonstrated a reliance on specific keywords and phrases for classifying a resume. For instance, GPT-4 categorized a resume as Information Technology based on the education in Computer Science, technical skills, and relevant work experience. This approach mirrors the tendency of our LLMs to focus on similar key indicators like educational background, technical skills, and practical experience.

The comparison highlights a common pattern across different models in resume screening, where certain terms and experiences significantly influence the classification outcome. Despite the similarities, one major difference between GPT-4's classification and the LMM's classification is that GPT-4 takes into account information more holistically allowing it to classify with better accuracy whereas the LLM's take in information as tokens and focus more on specific keywords. However, both showed a potential for biases due to their reliance on specific phrases and terms. Overall, this comparative analysis shows the importance of specific resume content in automated job category classification and the need to carefully tailor resumes to align with targeted job sectors.

5 Discussion

5.1 Limitations

While we were able to come to some valuable conclusions in our exploration of resume assessment and classification there are several large limitations that hamper our conclusions. First, based on our own literature study and other research, it is clear to us that while our model could very well be a piece of a resume assessment pipeline, we did not have the time nor the experience to fully replicate an entire resume assessment pipeline which in reality layer multiple machine learning and more classical NLP subsystems on top of each other to better parse, sort, classify, and rank resumes. Because our project only explored a subsystem of those larger systems, i.e. resume classification, we are limited in the overarching conclusions we can make about these systems as a whole. That being said, we can make conclusions about this vital component of these larger systems, and biases and flaws in one part of a system can easily propagate their way into the rest of a system. While this limitation prevents our study from being as comprehensive as we might like it does still provide a meaningful glimpse into the problems these systems very likely have. Secondly, while switching our model to Longformer did improve our issue with truncation, a more comprehensive solution is needed to process larger resumes holistically.

Another one of the larger limitations of our analysis comes from our dataset. While our dataset does contain a large number of resumes, these resumes were scraped from a website providing sample resumes for different jobs. Because of this, the resumes were pre-sanitized. In this process, all personal information like names, dates, company names, and locations were removed from the resumes and swapped with generic replacements like "city name". We theorize that this limits our ability to fully assess the biases that might arise in our fine-tuning process. It is not possible to discover if certain names might bias the classification system if the dataset we used does not contain any names to possibly bias the dataset. In retrospect, this sort of information obfuscation might be an effective way to prevent such biasing, but without a way to compare the effect with and without that information, we can make no conclusions. This limitation of the dataset also further removes our analysis from the reality of real resume classification systems, which are more likely to be trained on databases of real resumes, containing this kind of information.

5.2 Ethical Considerations

One of the driving forces behind our desire to explore this specific project is the ethical considerations found in this specific application of LLMs. While we were pleasantly surprised with how well our classification model behaved there are still some serious ethical questions the use of a model like this should bring up. Is it right to leave the fate of a person's livelihood up to a machine that is only 93% accurate? We know from relevant literature and our own experiments that it is very easy for LLMs to be biased by their training data. It is no great stretch to posit that classification systems like ours could easily be biased toward certain demographics, genders, or races if they are not trained carefully with biases in mind.

There are also a lot of ways these systems could easily be abused. Automated resume systems like these are in many ways black boxes. This would allow corporations to shift the blame from bad hiring practices and discriminatory behavior away from themselves and onto so-called 'errors' in their assessment software. Often we hear about transparency in hiring processes, but can there actually be any kind of real transparency if we don't know what these models are actually looking at to form their decisions?

Another consideration is how easy these models are to fool. In using a model like this without mitigating the issues of recency bias we found and the susceptibility to skewing the model with additional buzzword-laden text, a company would leave itself open to exploitation by bad actors. Nefarious individuals with the knowledge and power to exploit these vulnerabilities could use them to supplant applicants with more knowledge and experience. Furthermore, systems like ours could be used by attackers to fine-tune their own malicious text.

5.3 Extensions and Future Considerations

From our review of relevant literature, this specific area of research has many areas still to be explored. As resume assessment software continues to become more prevalent and start to involve more and more application of ML and LLMs the need for research into how these models think and what kind of biases might be affecting their decisions will only grow.

One way this project could be extended would be to build a larger resume assessment system based on some of the commercially available options. Adding other systems for parsing resumes into their component parts, extracting relevant skills, and ranking resumes in relation to a job description and each other would allow more extensive research to be done. It would also be interesting to try and collaborate with some of the companies that provide resume screening services to assess their systems for biases and other unethical or problematic behavior. In addition to this, implementing a system that handles larger resumes would be another worthwhile enhancement to this analysis. This could be accomplished in many ways, perhaps by using a different model or by breaking the resumes down into partitions and then analyzing each partition and summing the results.

Another future application of this technology could focus on helping job applicants. Currently, we mostly hear about resume assessment systems in terms of employers using them to screen applicants, but this technology could easily be applied to give applicants advice on how to tailor their resume towards specific job listings or give general advice on how to better present themselves on their resume. This is an application of the technology that we have not seen explored.

5.4 Replicability

Our overall experiments would be quite easy to repeat by any third party. We used a publicly available dataset which can be found on Kaggle and all of our finetuning was done on two open-source base models, Roberta, and Longformer. Both of these models are available to the public on HuggingFace.

Fine-tuning the models themselves was a bit difficult. Specifically the Longformer model is very large and at first, we ran into issues with too much

RAM utilization during the training process. However, using methods like gradient checkpointing we were able to train the model on Google Colab using a GPU with only 16GB of RAM. Anyone wishing to replicate our findings should be able to do the same. As for our visualizations, they were all created using SHAP, an open-source library with several good tutorials available online to guide a user through the process.

6 Conclusion

Our project, "Evaluating Resume Efficacy and Optimal Features Using LLM," revealed key insights into how automated resume-screening models function. We focused on how these models discern a resume's target job and their decision-making processes. Our findings showed that specific tokens and phrases significantly influence the model's decisions, highlighting how resume wording impacts its evaluation by automated systems. Our experiments demonstrated the model's vulnerability to manipulation with targeted text blocks, raising concerns about potential misuse. The project also showed the limitations and ethical issues of using LLMs in recruitment, including inherent biases in training data, imperfect accuracy, and a tendency to prioritize early text in a document. These aspects show the importance of transparency and accountability in automated recruitment processes.

Overall, our study contributes to understanding LLMs in recruitment, underlining the need for a balanced approach that acknowledges their benefits while being cautious of their limitations. As these technologies evolve, continuous scrutiny and refinement are important to ensure their effectiveness and fairness.

References

Amine Barrak. 2021. Toward a traceable, explainable, and fair jd/resume recommendation system.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Suhas Tangadle Gopalakrishna and Vijayaraghavan Varadharajan. 2019. Automated tool for resume classification using sementic analysis. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 10(1).

Brandon Grech and David Suda. 2020. A neural information retrieval approach for resume searching in a recruitment agency. *9th International Conference on Pattern Recognition Applications and Methods*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Arifa Sultana, S.M. Shawal Chowdhury, and Mrithika Chowdhury. 2023. Matching job circular with resume using different natural language processing based algorithms. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 490.