Project 2: Human-in-the-loop

Canfer Akbulut & Kerem Tuncer

I. Introduction

The age of rising automation makes it possible to render manual, time-intensive, and tedious processes - such as sorting through and evaluating hundreds of job descriptions - easy and instantaneous. Of course, machines cannot hope to parse through job descriptions with the level of nuance that humans can. However, a few well-chosen metrics on the part of the person designing the algorithm can allow for a process that achieves a healthy balance between convenience and quality.

Our task was to create an algorithm that allowed a person to input their resume and receive well-fitting yet varied job recommendations, based on 1,627 full-time, New York based position descriptions scraped from Indeed.com. This process is not fully automatic, but rather relies on a "human-in-the-loop" to assess the algorithm's performance at several critical points. What we hoped to do was to allow a person using our algorithm to receive three recommendations that were not only reasonably aligned with their qualifications and interests as stated by their resume, but also differed on a meaningful dimension, so as to avoid making three near-identical recommendations. Before we propose an approach to defining our metrics of goodness and diversity, it is essential for us to understand and visualize our corpus of job descriptions.

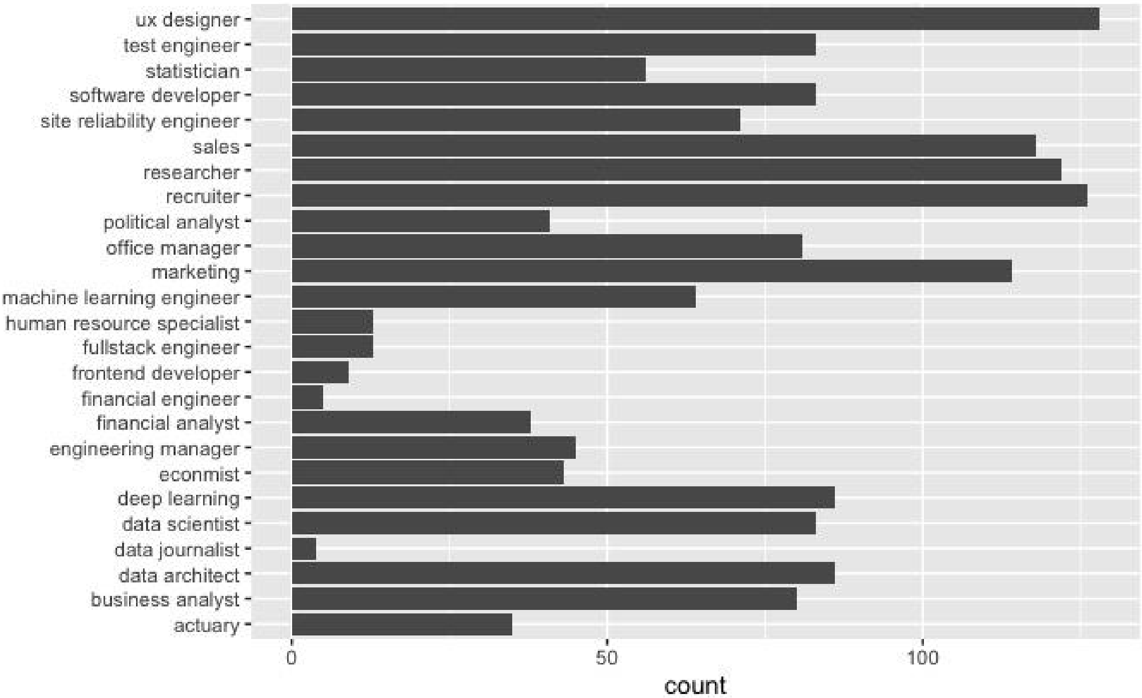


Figure 1: Bar graph of job role by count. Job role titles are provided by Indeed.com.

As is clear from the position titles in Figure 1, many of the job descriptions in our corpus are related in that they require a similar skillset. From a first glance, it seems that our algorithm will be most suited for use by candidates with a knowledge of mathematics, statistics, and programming. We must keep in mind the limitations introduced to our algorithm by our skill-specific dataset of job descriptions; a person without any statistical or computing background may receive recommendations that do not seem relevant to them because of the lack of variety inherent in our dataset. Still, we must strive to create a method broad enough to allow us to issue recommendations even to applicants from less directly-related fields.

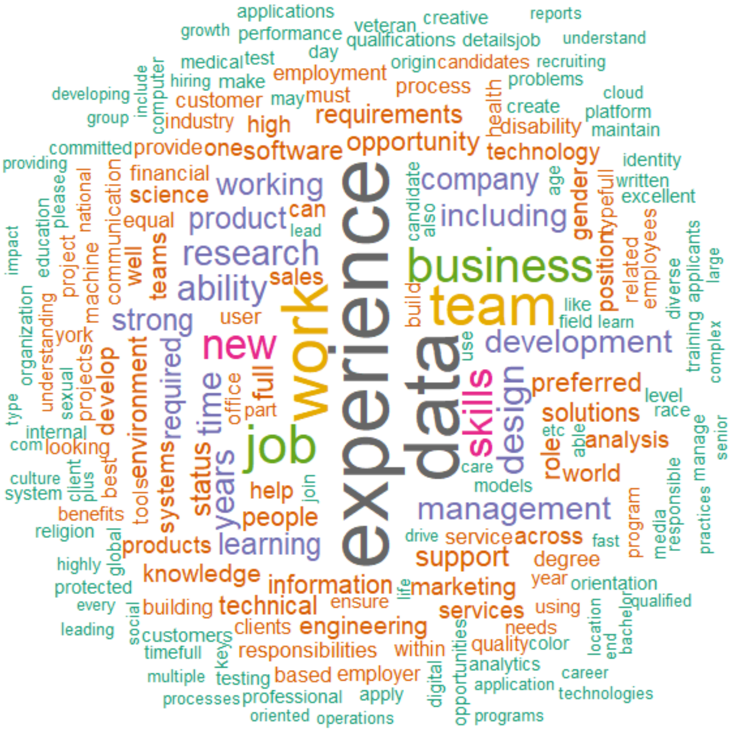


Figure 2: A word cloud plot of the most frequently used words in the Indeed.com job descriptions. The size of the words indicates the frequency of usage across job descriptions, with greater size corresponding to greater usage.

Some of the more frequently used words in Figure 2 are more informative than others. For instance, “experience” may not be a useful keyword, as a reasonable assumption that can be made before a job search is that all positions require some level of experience. Even “no experience required” positions will make use of this keyword. Others, however, give us a clearer view into our dataset of job descriptions. Skill-related words such as “software,” “management,” “analytics,” “engineering,” and “marketing” help us to determine the target audience of our algorithm, while commonly-used identity-related terms such as “veteran,” “gender,” and “disability” may allow applicants to assess their cultural fit with the companies whose vacant positions our algorithm recommends.

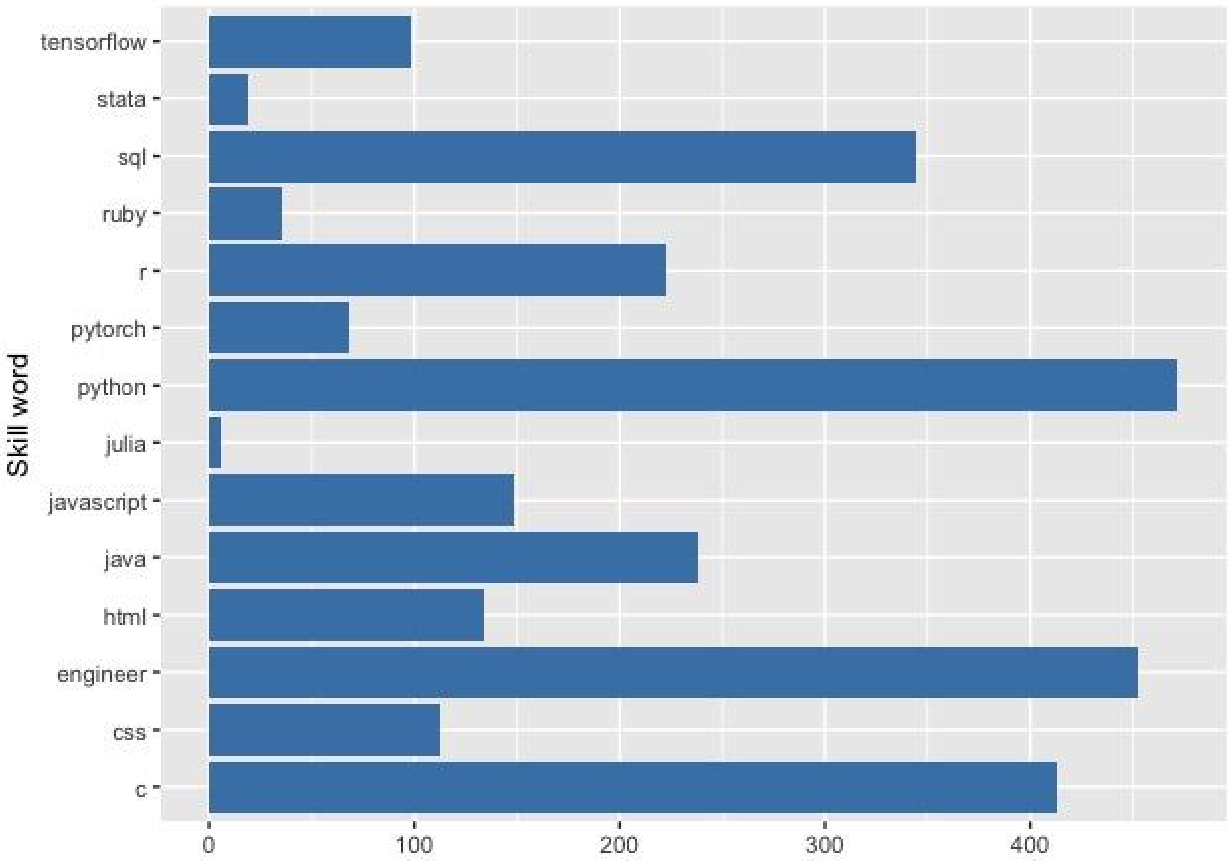


Figure 3. Bar plot of most frequently used skill words in the corpus of job descriptions. The list of relevant skill words was created by selecting keywords - primarily programming languages and software - from a collection of sample resumes.

Having been acquainted with our dataset, we are able to tackle the challenge of automating a significant part of the job-hunting process. Our goal is to create an algorithm that provides a user three job descriptions that sufficiently match the user’s resume, yet vary from one another on one or several meaningful and well-chosen metrics of diversity.

II. Pre-processing data

Standard text-cleaning procedures were applied to our dataset to ensure our algorithm could correctly interpret as much information as was available in the job descriptions. As per convention, we removed all capitalizations, punctuation, special characters, unicode encodings, and multibyte strings through regex commands. Then, we removed all stop words (using the base R *stopwords(“english”)* dictionary) and lemmatized all words (using the lemmatize_strings() function in the textstem package). Stop word removal allows for a less sparse tokenization matrix by limiting the amount of words that have no semantic value. Lemmatization achieves the same purpose by removing multiple versions of the same word, allowing us to treat words with the same root - like “experience,” “experiences,” and “experienced” - as functionally identical. We have also ‘squished’ all repeated whitespaces. Finally, job descriptions are tokenized so that each row corresponds to a job description, each column corresponds to a unique word across all job descriptions, and each value represents the frequency with which a word occurs within a description.

Raw job description	Clean job description
Agency :\nHavas Health & You\nJob Description Summary :\nN/A\nJob Description :\nN/A\nSkills :\nContract Type :\nPermanent\nHere at Havas across the group we pride ourselves on being committed to offering equal opportunities to all potential employees and have zero tolerance for discrimination. We are an equal opportunity employer and welcome applicants irrespective of age, sex, race, ethnicity, disability and other factors that have no bearing on an individual\x92s ability to perform their job.\nDue to high volume of applications, only eligible or matching candidates will be contacted by us.	agency havas health job description summary job description skill contract type permanent havas across group pride commit offer equal opportunity potential employee zero tolerance discrimination equal opportunity employer welcome applicant irrespective age sex race ethnicity disability factor bear individual ability perform job due high volume application eligible match candidate contact us

Figure 4. A sample job description in a raw and “clean” format. Note that text cleaning renders the job description shorter, allowing for a more populated tokenized matrix.

III. What is a “good” recommendation?

A. “Goodness” as a concept

Much of how we defined “goodness” had to do with what we were hoping to achieve with our semi-automated process of producing job recommendations.

A job description that is a “good fit” will ideally have a great amount of similarity with a user’s resume. How one chooses to operationally define similarity is also a matter of subjective judgment, but there are many well-validated methods of computing text similarity that can be applied to our scenario of matching job descriptions to users’ resumes.

There are several assumptions underlying our choice of an appropriate goodness metric. Firstly, we are assuming that the more relevant words a resume has in common with a job description, the greater the similarity between the documents. (Though there are several other ways to quantify similarity, we believe that using words as our units of comparison is the most straight-forward and purposeful approach to our dilemma.) Secondly, we are assuming that the greater the similarity between the resume and description, the better the fit of the description to the user’s skills, qualifications, and interests.

B. TF-IDF

As one of the most fundamental measures of term frequency in natural language processing, *term frequency-inverse document frequency* is an indicator of how relevant a word is to one document relative to a set of documents. It is computed by multiplying the frequency of a word in a document and multiplied by the inverse frequency of the word over a set of documents. What TF-IDF allows us to determine is whether the occurrence of a word in a document is informative or indicative of a certain trend or pattern, or whether we can discount its relevance to that document in particular. In the context of our dataset, a word like “experience” (recall its frequency as represented by Figures 1 and 2) has a very low TF-IDF value (0.00977). This indicates that the word “experience” may be common across many documents in the set; because it is so ubiquitous across job descriptions, its low TF-IDF statistic suggests that it has limited relevance to each individual job description. In comparison, words like “Python,” “R,” and “SQL” have TF-IDF indices of 0.04, 0.09, and 0.138, respectively. This indicates that when these words appeared within a document, they had greater relevance to the document than did words that were more frequently used across documents.

C. Cosine similarity

One of our selected metrics of similarity - and therefore goodness-of-fit - was cosine similarity. Cosine similarity is a metric that seeks to indicate how similar two documents are to each other. It seeks to remedy some of the most glaring shortcomings of Euclidean distance approaches to similarity; most notably, it calculates similarity without overweighting the importance of document size, a correction many other term frequency indices do not make. It makes this adjustment by taking into account only the orientation, not the magnitude, of a vector of words in multi-dimensional space. This means that even in circumstances where two vectors of words may have significant Euclidian distance between them as a result of varying document lengths, their orientations may still be closer together, indicating greater similarity. Since we are comparing documents of disparate sizes (an individual's resume tends to be much longer than the average job description on Indeed.com), we believed cosine similarity to be the metric best suited for our purposes.

Method: The resume text was appended to the vector containing the job descriptions. All the words in the corpus of job descriptions + resume were tokenized and vectorized using TF-IDF. Then, using the respective TF-IDF vector of each document as a weight, we computed pairwise cosine similarity scores between 0 and 1. For comparative purposes, we also computed pairwise cosine similarity scores without TF-IDF weighting.

D. Jaccard similarity

Less computationally complex than cosine similarity, Jaccard similarity defines the size of the union between two sets of items. In our context, an algorithm that seeks to compute a Jaccard index takes the resume of a user as one set and an Indeed.com job description as the other, then calculates the proportion of words that overlap across the resume and job description (for a visualization of the concept, see Figure 5). It provides a value between 0% and 100%, with higher values indicating greater similarity between the two text blocks.

Method: The resume text was appended to the vector containing the job descriptions. All the words in the corpus of job descriptions + resume were tokenized. Then, we computed pairwise Jaccard similarity scores between 0 and 1.

E. Latent Semantic Analysis

Finally, we decided to incorporate Latent Semantic Analysis (LSA) into our cosine similarity method that used TF-IDF weighting. LSA utilizes singular value decomposition, which is a method of decomposing a matrix into three matrices, to identify key concepts of a text. Its job gets difficult as there are certain words with several meanings. If you look at the top-middle part of our WordCloud figure, you will see the word ‘may,’ which has many meanings. To overcome this issue, LSA groups certain words together and compares them to how often they appear together in other text. In the end, the result is a LSA-scaling transformed matrix that can be used for similarity analysis.

Method: The resume text was appended to the vector containing the job descriptions. All the words in the corpus of job descriptions + resume were tokenized and vectorized using TF-IDF. Then, LSA-scaling transformation was applied to the document-term matrix that contained TF-IDF weights. Finally, we computed pairwise cosine similarity scores between 0 and 1.

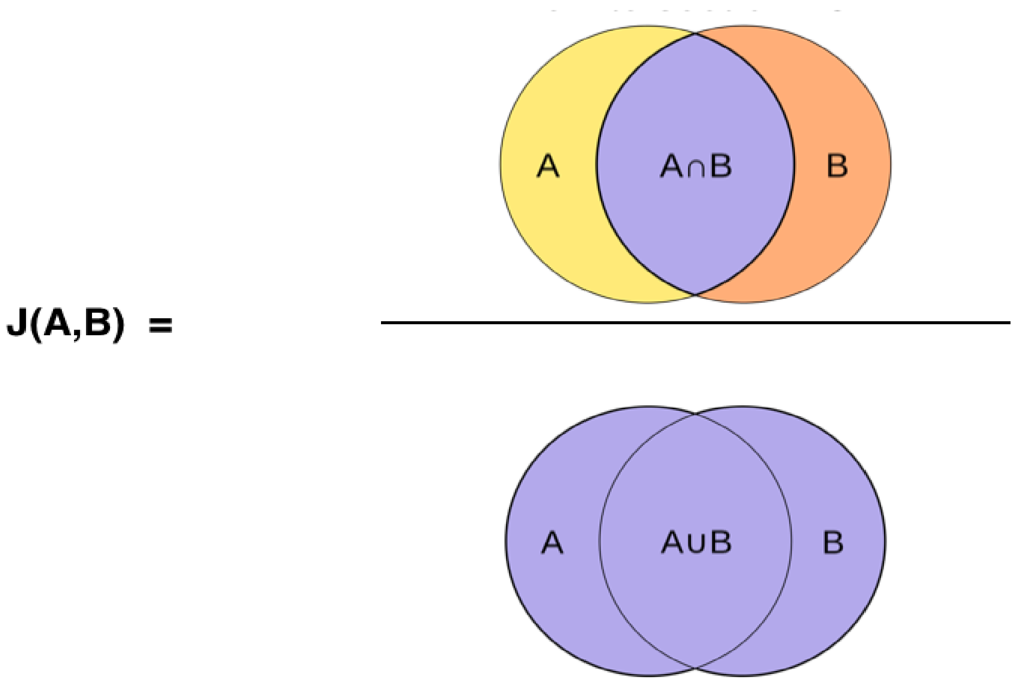


Figure 5. A visual representation of the Jaccard Index. Only the quantities highlighted in violet are computed in the equation for Jaccard Similarity. If A represents all words in a job description and B represents all words in a resume, then all co-occuring words across the two sets acts as the numerator. This value is divided by the count of all words in the job description and resume combined.

F. Limitations

Several limitations must be considered in evaluating the merit of both cosine and Jaccard similarity as metrics for goodness of fit. Though the general shortcomings of our algorithm in application will be discussed in the “Validation” section, a discussion of where cosine and Jaccard similarity may fall short as indices of “goodness” is in order.

Firstly, it is important to note that any similarity metric will be an oversimplification of reality. Occasionally, these metrics can mislead us; in our case, they may lead us to believe that a user’s resume and a job description are a better match than they truly are by prioritizing metrics that a human evaluator would be able to deem unimportant or irrelevant. For instance, “julia” is a keyword that appears infrequently relative to other words in our job descriptions. A resume that lists Julia as one of the user’s skills may result in an overinflated cosine similarity value, as cosine similarity assigns weight to how uncommon a word is throughout the sets of job descriptions. Especially in a scenario in which Julia is just one skill listed amongst many others, that cosine similarity is not an unbiased reflection of reality is a caveat one must consider carefully.

An important limitation of Jaccard similarity is that it can undershoot similarity if one or both of the documents is especially large, or if the sizes of the two documents are extremely disparate. As can be seen in Figure 5, the Jaccard index is sensitive to a change in word count for both of the documents, meaning that it is possible to bias this metric similarity in the negative or positive direction, depending on the total word count across the two documents. A short resume, for instance, may result in Jaccard similarity indices that are consistently lower than those computed on longer resumes, resulting in less accurate or more generic job recommendations.

IV. What are “diverse” recommendations?

A. “Diversity” as a concept

It is not sufficient for our algorithm to provide a user with three recommendations that are reasonably similar to their resume. We must also implement a feature that ensures the three recommendations vary from one another on at least one significant dimension. A primary motivation for this diversity metric is that we do not want to supply users with three identical job descriptions, as one of our objectives is to offer them a range of choices. While our implemented “goodness-of-fit” algorithm computes the similarity between the provided resume and the job descriptions to evaluate the best matches, the diversity metric needs to consider the differences *within* the job descriptions. Because comparing each job description to every other job description is an inefficient and resource-heavy process, we are using the “goodness-of-fit” index to significantly pare down the amount of job descriptions on which the diversity algorithm is run. That is, we are first computing how well the job description matches the resumes, then selecting the descriptions with a similarity value above a predetermined threshold to evaluate diversity within the descriptions.

B. Clustering attempts

Initially, we attempted to implement a diversity metric through clustering. Though this method failed to return a meaningful and reliable way of differentiating job descriptions, it is nonetheless a useful exercise to document our logic and highlight the shortcomings of this approach.

K-means clustering: What differentiates k-means clustering from other methods of unsupervised learning is that one specifies the amount of clusters to look for in advance. By defining a target number *k*, one is effectively determining the amount of centroids that the data will have, and consequently the number of "buckets" that are available for the machine learning algorithm to sort data points into. As we did not have a true k-value, we used the silhouette method to estimate the optimal number of clusters. This heuristic allows us to compute how similar each observation is to its own cluster relative to other ones. The best cluster amount is determined by assessing which k-value maximizes the average silhouette. We automated this process through fviz_nbclust() and got three as our k. Afterwards, we used our tf-idf matrix to divide our documents into three separate clusters. Unfortunately, the cluster sizes were worse than our expectations because the overwhelming majority of our descriptions were in a single cluster. We also tried the elbow method to see if a different k was necessary; however, the elbow method also did not produce any viable results. In the end, we went with k = 3.

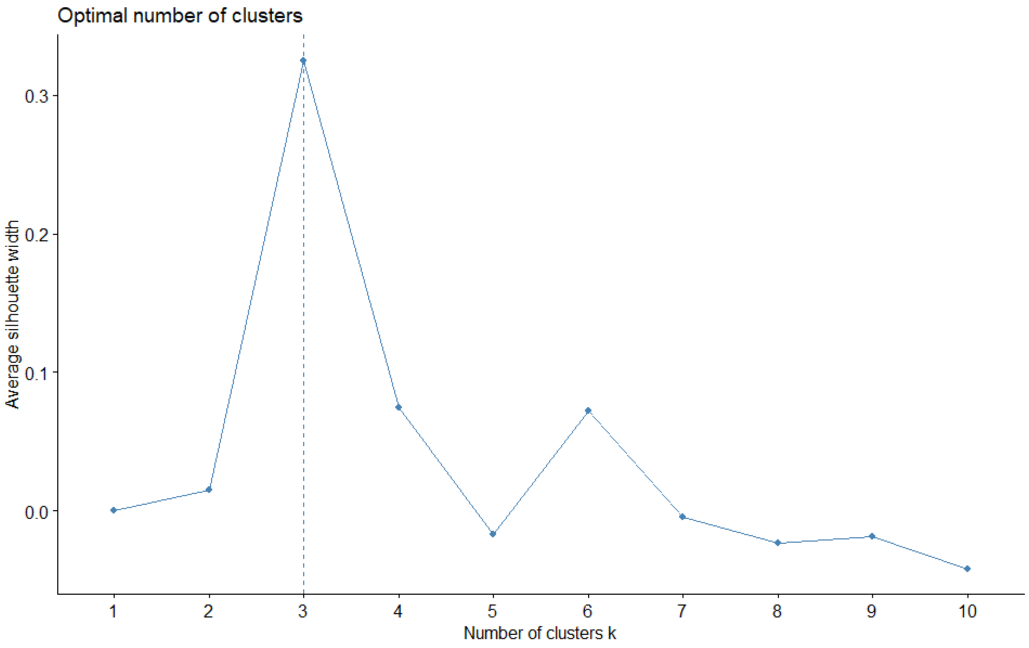
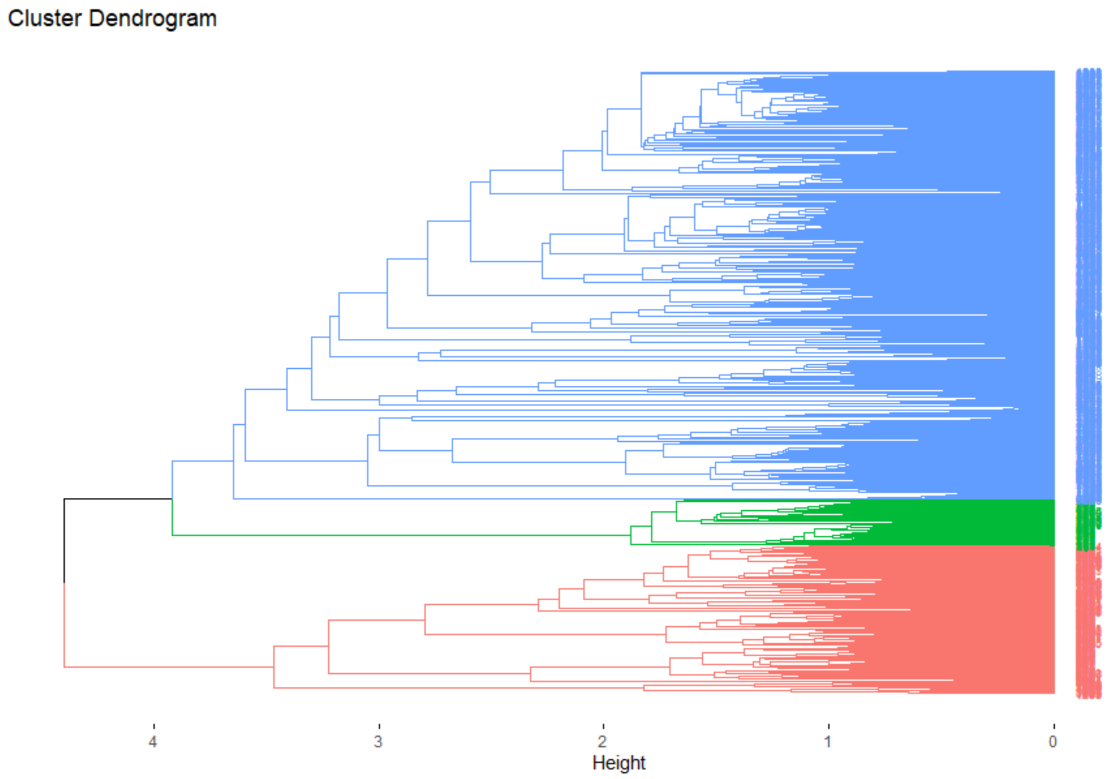


Figure 6. Silhouette plot representing the optimal number of cluster centers in our dataset.

Hierarchical clustering: Unlike k-means clustering, a hierarchical clustering process does not require the user to designate the amount of clusters before the fact. Instead, hierarchical clustering evaluates each data point as its own cluster, continuously assessing clusters with the greatest proximity and merging them into a single cluster. This is a significant advantage for our situation as we do not have a true number of clusters. Through an iterative process, hierarchical clustering is able to determine distances between clusters, which produce several relevant statistics. Single-linkage distance computation, for instance, provides us with an insight into the distance between two of the most similar clusters; complete-linkage indicates the distance between the two furthest, and therefore the two most dissimilar, clusters; average-linkage, on the other hand, signifies the distance between the centers of clusters. There is also the ward's minimum variance method that minimizes the total intra-cluster variance. We performed hierarchical clustering on four methods and examined their respective agglomerative coefficients. The highest AC was with the ward method, so we used it in our final hierarchical clustering model. Consequently, we cut the resulting "tree" into 3 clusters because we are supposed to produce 3 diverse resumes.

Figure 7. Dendrogram representing the hierarchical clustering of our dataset.



Though our application hierarchical clustering yielded three clusters, there were two reasons we were unable to apply hierarchical clustering as our metric of diversity. The first reason was that, though three clusters were ultimately returned, these clusters were highly unevenly distributed, as can be seen clearly in the dendrogram in Figure 7. Given our relatively limited dataset, we could not be confident that we would consistently find well-matching job descriptions from all three clusters. Additionally, upon exploring our clustering results, we found that there was no way for a human to distinguish between clusters. Though human judgment is subjective, our inability to determine why certain descriptions were grouped together indicated that hierarchical clustering did not provide results that were meaningful for our purposes.

Density clustering: This method of clustering leverages the belief that “a data space is a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density” (Sander, 2011). Density clustering makes fewer assumptions than other approaches to clustering, as it does not specify the variance or underlying density within groups in the data, nor does it necessitate a predetermined number of clusters as an input (Sander, 2011). Clusters formed out of density clustering may look different to clusters obtained through other methods, as density clustering is known to provide more “natural” clusters; that is, density clustering is not guaranteed to create clusters with within-cluster variance between the data points. More so than with other clustering methods, it can be difficult to detect discrete centroids through density-based clustering. This was to our detriment in our job-search application, as the implementation of this method was unable to consistently provide us with meaningfully different and unique clusters within our abridged job description set.

To do the clustering, we used the hbscan() function that requires the tf-idf matrix and the minimum points argument, which is the minimum size of a cluster. We set minimum size equal to the natural log of the sample size of job descriptions (one of several heuristic approaches). In the end, we had a total of 22 clusters. However, like our other clustering cases, one cluster had too many of the data points.

C. Methods - salary metric

Having exhausted unsupervised learning approaches to the diversity problem, we decided to try a more manual method of finding significant differences between the job descriptions. By familiarizing ourselves with the data set, we discovered that a significant amount of the job descriptions contained phrases or values that we could leverage to create a diversity metric. One of the most fruitful results was our attempt at creating a salary metric, or an indicator of compensation a user can expect to receive for every position. The advantage of this diversity metric was that its interpretation was more straightforward than metrics produced by unsupervised methods.

Method: We created a custom function through which we were able to extract salary information from all job descriptions that contained them. We achieved this by using regex to target the string "$," and extract the 3 words/numbers before the dollar sign and 2 words/numbers after it. Afterwards, we used the extract_salary() function from priceR library to automatically get the salary from our previous regex task. The advantage of using extract_salary() was that it gives the average value if the salary is a range (e.g. $50,000-$100,000 becomes $75,000).

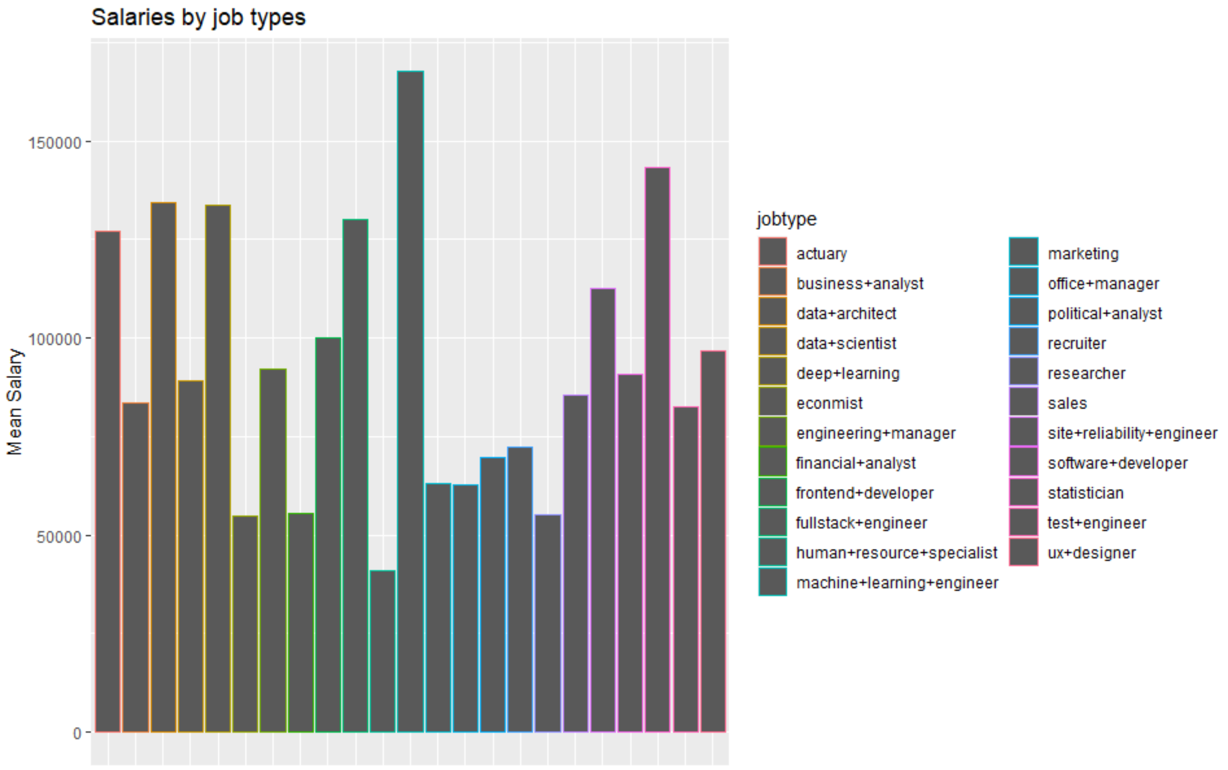


Figure 8. Mean salary by job title description.

D. Limitations

As discussed in previous sections, our attempts at clustering failed to return a meaningful metric of diversity upon qualitative evaluation. Our results for the salary metric are more encouraging, though the method presents its own unique challenges to interpretation. One limitation that we had to confront very early on in the process was that Indeed.com descriptions are not formatted identically, meaning that employers can choose what information to present and in which format. Some job descriptions may be lacking all information on salary; others may present salary information but in an unconventional format (ie.: without a dollar sign preceding the amount, shortening values in the thousands with a K, etc). Though we tried many variations of regex commands to increase the sensitivity of our targeting, there will inevitably be information we fail to extract - or information that is not available to be extracted - because of the lack of conventional formatting. We were able to extract reasonable salary values from a little over 50% of 1,627 job descriptions in our dataset. This limits the coverage of our diversity metric and potentially introduces bias into our measure of diversity, as it may lead to the exclusion of job descriptions with underlying commonalities. For instance, it is a possibility that unpaid positions do not include salary information, but we are not able to integrate this distinction meaningfully with limited information.

V. Validation

In order to validate our algorithm, we must test in application to several real-life resumes. As a stress-test for the various metrics in our algorithm, we will first feed in a resume and return job descriptions that are a poor fit according to our chosen dimensions. We will then use human judgment to qualitatively evaluate whether a job description that is indicated to be a poor fit is truly incompatible with the user's resume.

A. Resume pre-processing

All resumes were entered into the algorithm as txt files using the readLines() command. Standard text-cleaning procedures were applied to each line of the resume: all capital letters were changed to lowercase, spacing issues were resolved, punctuation marks and special characters deleted, stop words from the stopwords("english") dictionary removed, and all words lemmatized. Then, the lines were concatenated, outputting each resume as a single text string.

Raw resume line	Cleaned resume line
Assisting Prof. [*XYZ*] with his upcoming textbook on experimental research in social sciences. Creating R code for textbook exercises.	assist prof upcoming textbook experimental research social science create r code textbook exercise review chapter book content grammar

Reviewing chapters of the book for content and grammar improvement. Finding research papers to use as examples. Replicating and editing external datasets and R code.	improvement find research paper use example replicate edit external dataset r code

Figure 9. A sample line from an "average fit" resume used for validation in raw and clean format.

B. Bad Examples for Similarity and Diversity

To reproduce "bad" examples for similarity, we implemented the Jaccard similarity metric, TF-IDF weighted Cosine similarity metric, and the TF-IDF weighted + LSA-scaled Cosine similarity metric on our own resumes.

After computing the Jaccard scores on Kerem's resume, we sorted the values from lowest to highest. Although there were several job descriptions that had a zero match, we picked the one that was on the top after the sorting. The job description was for a UX Designer role that required knowledge of HTML & CSS for coding and Figma & Photoshop for visual design. The job primarily consists of creating mobile and desktop applications. However, Kerem is a student of Political Science and Statistics. He has no experience in UX design and he does not know any of the aforementioned software and programming languages.

We did the same procedure for TF-IDF weighted Cosine similarity. This time, the worst job description was even more irrelevant. It was for the Chief Medical Administrator position - a position that required a master's degree - at a medical clinic. This result has nothing in common with a Political Science-Stats resume.

Lastly, we found the worst-fitting job description using cosine similarity that was both weighted by TF-IDF and scaled through LSA. Like the other two trials, this computation also resulted in an unrelated job description. The role was for a test engineer role that required knowledge of IBM z/OS, Linux, and problem-solving on hardware and software. None of these qualifications are listed in my resume. All in all, we were happy to see that the worst matches using three different similarity methods produced expected results.

The same procedure was followed to find "bad" examples of job description matches for Canfer's resume. Using Jaccard similarity as our metric and sorting from lowest to highest, we found that the job that is most dissimilar to Canfer's resume is a position for a UX Designer for an online education service. Canfer is a Psychology major and Statistics concentrator with no prior UX/UI experience

who would not be prepared to “lead creation of user flows, wireframes, and prototypes.” The position also requires a BA/BS in Design, at least 4 years of experience with interactive design, as well as proficiency in HTML, Photoshop, Illustrator, iOS, and Android, none of which Canfer has.
Following a qualitative appraisal, we can conclude that this position is a poor fit for Canfer, which is what is indicated by our Jaccard similarity metric.

Using TF-IDF weighted cosine similarity, we find that the position that least matches Canfer’s resume is an office manager job for “a very posh concierge service.” Though this position is shorter than most others in our dataset, providing us with limited information to base our evaluation on, Canfer does not have the required 1-year experience in management and medical experience, nor is she interested in a position as an office manager. Using cosine similarity scaled through LSA and weighted through TF-idf, the worst match for Canfer’s set of qualifications and interests is a position as a Hardware Developer at a multinational technology company. This job is unsuitable for Canfer on several counts: she does not hold a BS in electrical engineering, mechanical engineering, or the physical sciences; she does not have 3+ years of experience in semiconductor tests; she has no understanding of “microprocessor circuit characterization,” nor does she have experience with ATE equipment or circuit board design. It seems that all three of our chosen “goodness-of-fit” metrics are adept at finding descriptions that have little relevance to a user’s resume.

We also wanted to demonstrate that our diversity metric of salary filters out job descriptions that are good but not diverse. For example, Kerem’s resume's second strongest match under TF-IDF + cosine method was another quantitative political scientist position at a university. This description did not have a salary tag just like the description that had the highest similarity score under this method. Interestingly, it also described a very similar position. Therefore, our diversity metric successfully deprioritized this job description.

C. Good Examples for Similarity

To get our good examples, we used the Jaccard similarity, TF-IDF weighted Cosine similarity, and the TF-IDF weighted + LSA-scaled Cosine similarity on our own resumes.

For the Jaccard Similarity, Kerem’s resume was matched with a Data Scientist role that had a rather short job description. Given his Political Science-Stats major, a majority of Kerem’s professional experience has been in internships and research assistantships involving data science. Hence, this is a decent match.

Then, the TF-IDF weighted cosine similarity result for Kerem was a perfect fit. The Department of Political Science at NYU was looking for a quantitative political scientist to join the Social Media and Political Participation Lab. Kerem has several internships where he worked with social media on political topics. The description had Python, R, and SQL as the required computing languages, which are all included in his resume. The work involved machine learning, natural language processing on political data. Most of Kerem's experiences have been at the intersection of these two. Hence, these details fit Kerem's experience very well. Honestly, this is a dream job for Kerem.

Consequently, we found the best-fitting job description using cosine similarity that was both weighted by TF-IDF and scaled through LSA. It was for a statistical consulting role at Cornell University's Statistical Consulting Unit. The role required a degree in Statistics and it consisted mainly of assisting people in Cornell's departments, such as social sciences ones, with statistical analyses. Like the job description for Jaccard similarity, this is also an opportunity that Kerem would have been interested in if he had graduated. All in all, this method and the Jaccard similarity resulted in expected job descriptions, whereas the TF-IDF weighted cosine similarity did not perform too well.

Following the same procedure for Canfer's resume, using a Jaccard similarity index to pinpoint the best match for Canfer returned a research position at a trading firm. It is worth noting that this job description was among the shortest in our dataset, and without normalizing for word count, the length of the description may have affected the performance of our algorithm. Some of Canfer's experience, such as her knowledge of Python and R, may qualify her for this role. Our TF-IDF-weighted cosine similarity metric returned a position as a statistician at an R&D laboratory environment. Despite requiring an advanced degree - which Canfer does not yet possess - Canfer's knowledge of R, SAS, linear and nonlinear regression, mixed-effects models, and experimental design (as listed on her resume) suggest to the algorithm that she may be a good match for the position.The best-fitting job description through our TF-IDF weighted and LSA-scaled cosine similarity was a research position at a hospital. Though the exact research topic does not coincide with Canfer's research interests, with plenty of experience in laboratory settings and some medical imaging experience, this position could be a good fit for Canfer in the future.

D. Good Similarity & Diversity

To assess diversity in our dataset, we first computed all three of the goodness metrics discussed above. Then, we selected the best-performing goodness metric for each of our levels of diversity. The three levels that we were able to extract were 1) no salary information, 2) below $80,000 average salary, and 3) above $80,000 average salary. This means that for a user's resume, each metric returns three of the top-performing job descriptions based on similarity, all of which differ on a chosen significant dimension: indicated salary level.

For Canfer's resume, the three suggestions that were returned through the Jaccard similarity metric were 1) a researcher at a trading firm (0.19 and no salary tag), 2) an administrator at an insurance firm (0.11, <$80k), 3) and a test engineer at a technology corporation (0.11, >$80k). These three were diverse on the salary metric, as the first is paid above $80,000, the other is paid below $80,000, and the third contained no meaningful indication of a salary per annum. Other than our primary diversity metric, these jobs were also diverse in their requirements, scope, and field.

The three descriptions returned using a TF-IDF weighted cosine similarity metric were 1) a statistician at an R&D laboratory (0.14, no salary tag), 2) a data scientist at an ocean shipping company (0.10, <$80k), and 3) a partnership development manager at a digital mental health company (0.08, >$80k). Once more, these three job descriptions fulfilled our criteria of diversity based on salary, but also offered diversity in terms of the skills required of the applicant and topics explored within the field. The first two jobs are most compatible with Canfer's experience with data analysis and statistics, while the third may leverage her interest in mental health, her experience in mental health advocacy, and her primary field of study: psychology.

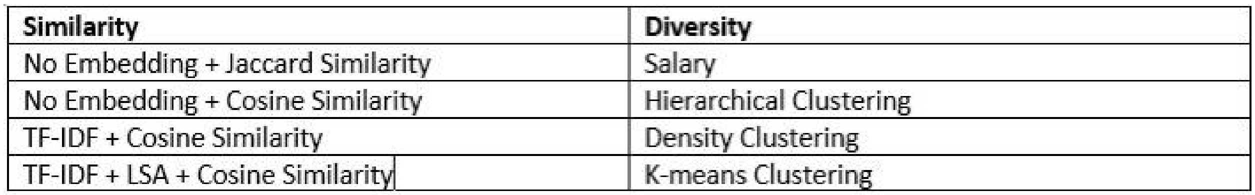
The three descriptions returned using a TF-IDF weighted and LSA-scaled cosine similarity metric were 1) a researcher in a hospital setting (0.99, no salary tag), 2) an administrator at a shelter for adults (0.99, <$80k), and 3) a business office manager in a nursing facility (0.99, >$80k). While this metric fulfilled our criterion of diversity on the salary dimension, several of these positions are similar in that they are administrative roles in a people-facing role. Given Canfer's administrative experience in laboratory settings, these positions are not incongruous with her resume; however, other methods of evaluating goodness-of-fit coupled with the salary diversity metric tended to produce recommendations that were more varied in scope.

For Kerem's resume, the three suggestions that were returned through the Jaccard similarity metric were 1) a data scientist at an advertising firm (0.29, no salary tag), 2) an office manager at a dental office (0.11, <$80k), 3) and a test engineer at a technology corporation (0.11, <$80k). In the process, we followed the exact same diversity metric that we used for Canfer's resume [no salary tag, <$80k, >$80k]. The first description fit Kerem's background given his data science internships. However, the other two roles were disappointing. His qualifications did not fit the test engineering position (except for Selenium knowledge) and the other position did not have any functions that he would be interested in doing.

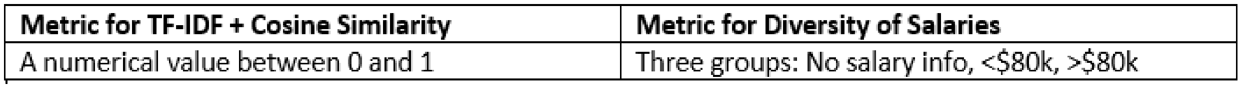
The three descriptions returned using a TF-IDF weighted cosine similarity metric were 1) quantitative political scientist at NYU (0.13, no salary tag), 2) a consultant at a political consulting firm focusing on election campaigns (0.12, >$80k) , and 3) a quantitative political analyst at a policy institute (0.09, <$80k). As we have previously discussed, Kerem's most recent research assistantships at Columbia have combined political science and data science, which fit the first and the third description. Likewise, Kerem has interned with two different legislative offices, where he also helped with their election campaigns, which fits the second description. In addition to salary diversity, there was a diversity captured in quantitative (1st and 3rd) vs. qualitative (2nd) positions.

The three descriptions returned using a TF-IDF weighted and LSA-scaled cosine similarity metric were 1) a statistician at Cornell's Statistical Consulting Unit (0.99, no salary tag), 2) a quantitative political analyst at a policy institute (0.99, <$80k - same job as the last one in the previous group), and 3) a risk management analyst at a financial institution (0.99, >$80k). The first two descriptions did fit Kerem's background; however, the last one did not. Interestingly, risk management is a field that Kerem is interested in pursuing in the future. But his current resume did not include any elements that fit with the requirements and the qualifications of that description.

IV. Conclusion



Similarity	Diversity
No Embedding + Jaccard Similarity	Salary
No Embedding + Cosine Similarity	Hierarchical Clustering
TF-IDF + Cosine Similarity	Density Clustering
TF-IDF + LSA + Cosine Similarity	K-means Clustering



Metric for TF-IDF + Cosine Similarity	Metric for Diversity of Salaries
A numerical value between 0 and 1	Three groups: No salary info, <$80k, >$80k

Figure 10. A visual summary of the methods and metrics that were tried in the process. We found the best performance using TF-IDF weighted cosine similarity for goodness-of-fit and a salary metric for diversity.

In devising a human-in-the-loop process that automated the search for job descriptions in our dataset, we applied a variety of metrics to assess both goodness and similarity. Some metrics were more effective than others in producing job recommendations that both adequately addressed the similarity between a user's resume and the description and demonstrated significant and meaningful variation within the three recommendations. Upon evaluation, we have found that using TF-IDF weighted cosine similarity in conjunction with our aforementioned salary diversity metric has yielded the most well-fitting and varied results. However, the Jaccard similarity method and the LSA-scaling method also resulted in relatively valuable recommendations, as well. In the future, it would be interesting to assess the performance of our algorithm on a less homogenous dataset of job descriptions. Doing so may spotlight more structural weaknesses than we were able to pinpoint with our current application. For instance, a specific improvement we would implement in a future edition would be a way to filter the recommendations by preferred education level. Even job descriptions fitting our resumes had degree expectations that were above a bachelor's degree. Though, we are confident that our model is not overly tailored for our resumes because the similarity method we used does not allow that. A large-scale application of our algorithm would require much more fine-tuning, but in its current state, it performs well under the specifications of the assignment.