Ordinal Regression for Job Search Keyword Similarity Prediction

Md Ahsanul Kabir[1], **Kareem Abdelfatah**[1], **Mohammed Korayem**[1], **Mohammad Al Hasan**[2]
[1]CareerBuilder LLC, [2]Indiana University Indianapolis, USA
{mdahsanul.kabir, kareem.abdelfatah, mohammed.korayem}@careerbuilder.com, alhasan@iu.edu

Abstract

Job search platforms face growing challenges in efficiently matching job seekers with relevant opportunities, making Job Search Keywords (JSK) similarity prediction a critical task. Within the scope of this task, the prediction algorithms must accurately evaluate the relevance of user-provided keywords, such as job titles, skills, and industries. Also traditional regression models for predicting similarity scores often treat all errors equally. In practice, mistaknes within high-similarity ranges (e.g., "software engineer" vs. "developer") can severely impact user experience, whereas errors in moderate similarity ranges (e.g., "data scientist" vs. "data analyst") are comparatively less disruptive. Addressing this issue requires a model that prioritizes accurate predictions for highly similar JSK pairs.

In this paper, we present an ordinal regression framework tailored for predicting JSK similarity. We propose a custom loss function that penalizes errors based on their ordinal label distance, ensuring higher accuracy for high-similarity matches. Through extensive experiments with multiple baseline models for ordinal regression on job data, we demonstrate that our approach, utilizing the All Threshold Functional Error Loss (ATFES) function, achieves the best results based on multiple evaluation metrics, such as, Mean Absolute Label Error (MALE) and Root Mean Squared Label Error (RMSLE).

Introduction

Accurately predicting semantic similarity between a pair of job search keywords (JSKs) is a critical task for improving user experience in online job platforms. Similar to other online platforms, such as, e-commerce (Fuchs et al. 2020), and Web search (Ramalingam et al. 2024), job search platforms also rely on job search similarity prediction for building various assistive applications in their platforms; examples include "query expansion,", "recovery from zero recall," and "query rewriting." These applications in online job platforms help the job seekers to discover variety of relevant positions and help the recruiters to identify suitable candidates efficiently. However, query similarity prediction in the job search domain presents unique challenges due to the domain-specific nature of JSKs, which often include specialized terms, industry jargon, or skill-based expressions. So, keywords similarity prediction for the job search domain needs substantial research, which is the focus of this work.

Traditional information retrieval methods that perform well on typical web documents often struggle with short, unstructured, or highly contextualized queries, as observed in e-commerce search (Mandal, Khan, and Kumar 2019; Sun et al. 2021). While token-level similarity provides a simple and interpretable approach (Qi, Wu, and Mamoulis 2016), it is prone to false positives and lacks the capability to capture deeper semantic relationships. For instance, in the job search domain, two queries such as "software engineer Java" and "Java backend developer" might have different token representations but are semantically similar due to overlapping skill requirements. To overcome this issue, contextual information, which provides a richer context to compare two job queries, such as, skills associated to a job, or the specific industry sector to which the job belongs to, need to be incorporated in the similarity prediction model.

Another critical consideration is the distribution of similarity values and the varying impact of prediction errors across the ranges of similarity values. As similarity value, a positive real number between 0 and 1 is returned by most of the job similarity prediction models. These models use a regression setup with L_2 loss function, which provides uniform attention over the entire range of similarity values as they predict (Rennie and Srebro 2005). But in real-life usage of such models' prediction, mistakes (both overestimating and underestimating) in high similarity values (e.g., above 0.85) are particularly catastrophic, as such job keyword pairs are often used for JSK expansion, job recommendation, and candidate matching. Incorrect predictions for highly similar JSKs thereby lead to suboptimal recommendations and degraded user experience. To overcome this limitation, ordinal regression approaches should be used which enables the model to prioritize the correct performance of high-similarity value predictions (Cao, Mirjalili, and Raschka 2020).

In this work, we propose a novel supervised learning framework that formulates JSK similarity prediction as an ordinal regression task, introducing three key contributions. First, we transform continuous similarity values into ordinal buckets, drawing inspiration from a previous research work in e-commerce, using finer granularity for highly similar JSKs. Second, we design a custom loss function that prioritizes accuracy in the highest similarity bucket by penalizing errors more heavily in this region, without compromising overall performance. Third, we leverage advanced feature extraction techniques, such as BERT embeddings (Devlin et al. 2018) and spherical embeddings (Meng et al. 2019), to represent queries in a semantically rich vector space. Our results underscore the importance of integrating domain-specific features and focusing on high-similarity predictions to improve overall performance of job query similarity tasks.

Problem Formulation

Let $\mathcal{D} = \{(q_i, q_j), s_{ij}\}_{i=1}^{N}$ be a dataset containing pairs of JSKs (q_i, q_j). Each JSK, q_i is associated with a list of skills denoted by $\mathcal{S}_i = \{s_1^i, s_2^i, \ldots, s_m^i\}$, and similarly, q_j is associated with a set of skills $\mathcal{S}_j = \{s_1^j, s_2^j, \ldots, s_n^j\}$. In addition to skills, *carotenes* (denoted as C_i for q_i and C_j for q_j) represent taxonomy nodes associated with the respective JSKs. Each pair (q_i, q_j) in the dataset has a similarity score $s_{ij} \in [0, 1]$ that quantifies the semantic overlap between the two JSKs, where a score of 1 indicates perfect similarity. The goal of the supervised learning task is to train a model that predicts the similarity value $\hat{s}_{ij}$ for unseen JSK pairs. Optionally, the model can incorporate the skills $(\mathcal{S}_i, \mathcal{S}_j)$ or carotenes (C_i, C_j) during the representation learning process to enhance prediction performance.

Initially, similarity scores are derived using a combination of user signals, cosine similarities from multiple embedding methods, and other features. However, predicting continuous similarity values directly with regression models presents challenges. A major issue is that traditional regression assigns equal penalties to prediction errors across the entire similarity range, which may not align with the nuanced requirements of JSK similarity prediction tasks. Specifically, prediction errors are more critical for highly similar JSKs than for moderately or weakly related ones.

For example, JSK pairs with high similarity scores often represent identical or nearly identical concepts, making them ideal for tasks like JSK expansion or reformulation. Incorrect predictions for these pairs can result in poor JSK suggestions and negatively affect user experience. On the other hand, small deviations in similarity scores for less related JSKs have minimal impact. Therefore, the learning objective must emphasize greater accuracy in predicting higher similarity scores, where the cost of errors is more significant.

To address this, continuous similarity scores are transformed into discrete ordinal buckets, with finer granularity at higher similarity levels and coarser granularity at lower levels. This ensures the model prioritizes ranking accuracy for highly similar JSKs without adding unnecessary complexity for less related ones. Each bucket corresponds to a range of similarity values, with higher ordinal labels indicating stronger semantic overlap. We define a mapping function $H : [0, 1] \to \{0, 1, 2, 3, 4\}$, which assigns each similarity score to one of five ordinal buckets. For instance:

$$H(s_{ij}) = \begin{cases} 0 & \text{if } 0 \le s_{ij} \le 0.14 \\ 1 & \text{if } 0.14 < s_{ij} \le 0.44 \\ 2 & \text{if } 0.44 < s_{ij} \le 0.65 \\ 3 & \text{if } 0.65 < s_{ij} \le 0.85 \\ 4 & \text{if } 0.85 < s_{ij} \le 1.0 \end{cases} \tag{1}$$

The thresholds are designed to evenly distribute the in-house master dataset across the buckets. Each ordinal label holds a distinct meaning: pairs labeled 4 represent the most similar JSKs (scores above 0.85), which are the highest priority. Labels 3 and 1 correspond to similar and dissimilar JSKs, respectively, while label 2 denotes neutral pairs that are neither clearly similar nor dissimilar. Label 0 represents completely unrelated JSK pairs.

Methodology

We implement a neural network-based model for performing ordinal regression to predict JSK similarity. Our model's output layer is designed to accommodate an ordinal loss function, which is described in detail below, along with the architecture of the model.

Loss Function

Previous research, such as ORDSIM (Kabir et al. 2022), utilized a loss function that inadequately addressed the differences between ordinal labels, treating all labels uniformly. The ORDSIM loss function is expressed as:

$$ATMSEL = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - \lambda_l)^2 \tag{2}$$

ORDSIM manually adjusted thresholds to account for the uneven weighting of ordinal labels. However, this approach only partially mitigates the issue and often leads to a centralization effect, similar to other L_1 or L_2 loss functions, where the predictions tend to converge toward the mean.

To improve upon this, we propose assigning distinct weights to different ordinal labels. Selecting appropriate weights poses challenges, as the values may introduce non-uniformity. Moreover, while minimizing centralization is crucial, the model's performance metrics—such as Mean Absolute Label Error and Mean Squared Label Error—must also remain competitive.

We address these concerns by defining a slow-moving exponential weight function that generates smooth and appropriate weight values. Although we experimented with other functions, the proposed All Threshold Functional Error Loss (ATFES) yielded the best results. The ATFES loss function is given by:

$$ATFES = \frac{1}{N} \sum_{i=1}^{N} f_n(l)(\hat{y}_i - \lambda_l)^2 \tag{3}$$

$$f_n(l) = 0.01e^l + 1.0 \tag{4}$$

Here, $f_n(l)$ represents the weight associated with the ordinal label l. As l increases, the weight also increases.

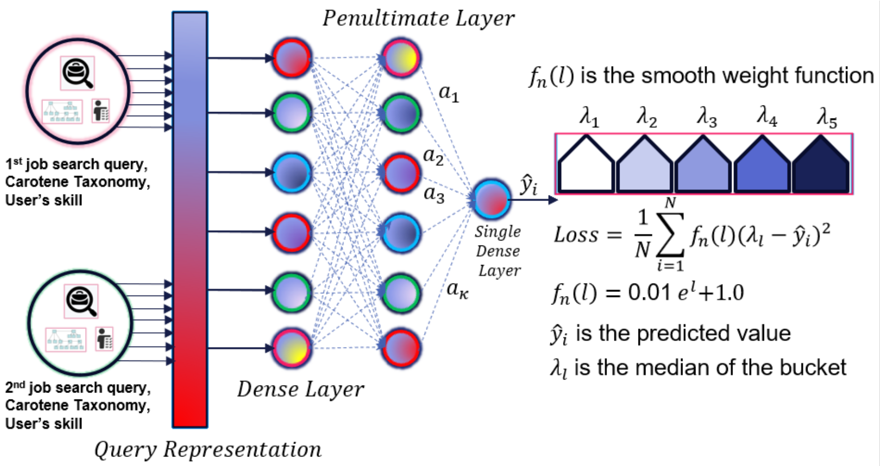

Penultimate Layer
$f_n(l)$ is the smooth weight function
a_1
λ_1 λ_2 λ_3 λ_4 λ_5
a_2
a_3
$\hat{y}_i$
1st job search query, Carotene Taxonomy, User's skill
Single Dense Layer
$Loss = \frac{1}{N}\sum_{i=1}^{N} f_n(l)(\lambda_l - \hat{y}_i)^2$
a_κ
$f_n(l) = 0.01\, e^l + 1.0$
$\hat{y}_i$ is the predicted value
λ_l is the median of the bucket
2nd job search query, Carotene Taxonomy, User's skill
Dense Layer
Query Representation

Figure 1: Architecture of the Ordinal Regression Model with Smooth Weight Function Achieving Optimal Performance

Specifically, for $l \in [0, 1, 2, 3, 4]$, the corresponding weights $f_n(l)$ are $[1.0, 1.03, 1.07, 1.2, 1.5]$. This property ensures that higher ordinal labels receive greater emphasis, thereby reducing the centralization tendency and prioritizing critical similarity buckets effectively.

Training Procedure and Model Architecture

To optimize our model for the ordinal regression task, we utilize the ATFES Loss applied at the output layer of a perceptron, leveraging a mini-batch training strategy. Each training sample $x_i = (q_1, q_2)$ from the dataset serves as input, where the JSKs q_1 and q_2 undergo a representation learning process before being passed through a neural network comprising two dense layers.

In this study, we select the top five skills for each q_i, separated by commas. Furthermore, we use only the carotene text for the taxonomy nodes, as prior research observed that the graph structure within the taxonomy nodes showed limited utility due to mismatches in representation space (Kabir et al. 2022). For the best-performing model, we adopt a feature extraction technique inspired by recent work to achieve a simpler yet effective representation. Specifically, all features are treated as text features and concatenated (Kabir et al. 2024).

Let $a_1, a_2, \ldots, a_{i_\kappa}$ denote the outputs from the penultimate layer for the input x_i, with corresponding weights $w_1, w_2, \ldots, w_\tau$. The predicted similarity value $\hat{y}_i$ is then computed as:

$$\hat{y}_i = \sum_{j=1}^{\tau} a_j \times w_j \tag{5}$$

Consider a training instance with a true similarity value of 0.95, which falls in the $(0.85, 1.0]$ bucket centered at $\lambda_l = 0.925$. If the predicted value $\hat{y}_i$ is 0.901, the weight of the corresponding ordinal label l is 1.5, derived from the weight function $f_n(l)$. ATFES penalizes the squared deviation from λ_l, producing an error term of $1.5 \times (0.901 - 0.925)^2$. By minimizing these errors, ATMSEL encourages predictions to converge toward the bucket medians with appropriate weighting, thereby capturing JSK similarity more effectively than traditional methods.

To illustrate the weight function's impact, consider an example with ordinal labels 0, 1, 2, 3, and 4. If the predicted label consistently averages to 2.0, the Mean Absolute Label Error (MALE) would be 1.2, indicating a low error. This demonstrates a centralizing tendency in predictions when using ATMSEL loss. However, with ATFES loss, the calculations yield:

$$[1.0, 1.03, 1.07, 1.2, 1.5] \times [2, 1, 0, 1, 2] = [2.0, 1.03, 0.0, 1.2, 3.0] \tag{6}$$

The average of this result is 1.5, meaning the ATFES loss reduces the centralization tendency by 25%.

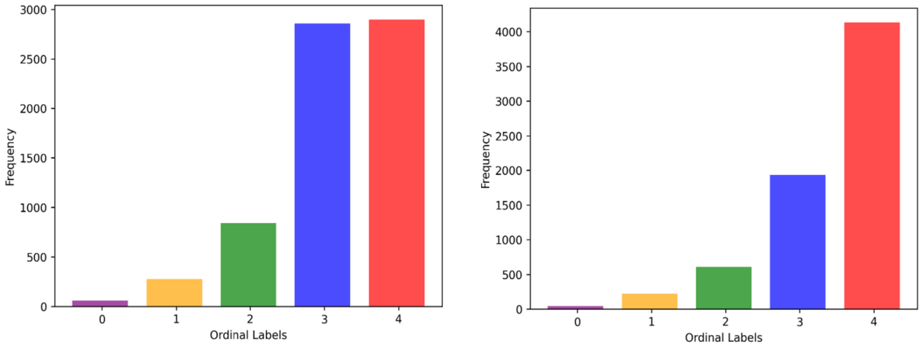
The architecture, depicted in Figure 1, showcases our main model, which remains consistent across both ORDSIM and our proposed framework. Both models start with input embeddings for q_1 and q_2, enriched with JSK text and category path information. These embeddings feed into two hidden layers, configured with N_κ and N_τ neurons and dropout probabilities of p_1 and p_2, respectively. Each hidden layer uses the ReLU activation function.

Dataset

The master dataset used in our study consists of 175K instances and includes the following columns: the first JSK, job skill, and carotene category for the first JSK, as well as the second search JSK, job skill, and carotene category for the second JSK. Additionally, the dataset contains a similarity score between the two JSKs, which we convert into ordinal buckets. We split the master dataset into training, testing, and validation sets with a 6:2:2 ratio.

Baseline Methods

Our approach is compared against several baseline methods, including CORAL-Ordinal (Cao, Mirjalili, and Raschka 2020), CORN (Shi, Cao, and Raschka 2021), OLL-1 (Castagnos, Mihelich, and Dognin 2022), and a Scaled-



(a) Predicted Ordinal Label Distribution with ORDSIM

(b) Predicted Ordinal Label Distribution with ATFES (best)

Figure 2: Predicted Ordinal Label Distribution for Most Similar Pairs (Ordinal Label = 4)

Table 1: Comparison with Baseline Methods for all the representations

Baseline Methods	Embedding	AllFeaturesUsed	MALE	RMSLE
CORAL-Ordinal	BERT	Yes	1.07	1.39
CORN	BERT	Yes	1.03	1.35
ORDSIM	BERT	Yes	0.668	0.894
OLL-1	BERT	Yes	0.723	0.898
Scaled-Loss	BERT	Yes	0.781	1.04
ATFES	In-House	Yes	0.898	1.14
ATFES	Spherical	Yes	0.811	1.05
ATFES	BERT	No	0.702	0.93
ATFES	**BERT**	**Yes**	**0.662**	**0.881**

		Predicted				
		0	**1**	**2**	**3**	**4**
	0	4603	1573	744	312	67
	1	1198	2779	1416	684	154
Actual	**2**	423	1223	2903	2088	762
	3	186	674	1590	3467	1553
	4	40	233	605	1930	4133

Table 2: Confusion Matrix for Ordinal Regression Prediction with ATFES (Best)

Loss based loss function. These baseline methods are capable of performing ordinal regression on JSKs. Each method employs the `bert-base-uncased` pre-trained BERT model for feature extraction, ensuring consistent criteria across models. The primary difference among these models lies in the loss function used. While we apply the specified loss functions and hyperparameters for all baseline methods, the Scaled-Loss baseline represents a unique model we designed. In this model, we implemented a linear weighting function in the loss calculation, as opposed to an exponential loss function. The linear transformation is defined as:

$$f_n(l) = e/10.0 + 1.0 \tag{7}$$

The concept behind this design is to use a linear weight function, rather than relying on an exponential weighting approach.

Hyperparameter Tuning

We have multiple versions of our model where we keep the loss function similar, but the representation of the text is different. In one configuration we use the in-house embedding of our company which is basically CNN based embedding. In another version we use Spherical Text Embedding (Meng et al. 2019).For both these versions the representation of the words are used through a embedding layer. We use number of epochs is 1000, with early stopping. For the other two versions we use BERT pretrained model (Devlin et al. 2018) to perform the Ordinal Regression task. We integrate the custom ATFES loss function for all the versions. The BERT model is tuned for 1000 epochs using early stopping and default learning rate.

Evaluation Metrics

We calculate two evaluation metrics. Both of the metrics is calculated based on the predicted Ordinal Labels. The idea is to first transform the predicted value into Ordinal Label using the thresholds we illustrated earlier. The two evaluation metrics we use here are Mean Absolute Label Error (MALE) (Kabir et al. 2022) and Root Mean Squared Label Error (RMSLE) which are defined by the following equations.

$$MALE = \frac{1}{N}\sum_{i=1}^{N} |\hat{l}_i - l_i| \tag{8}$$

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\hat{l}_i - l_i)^2} \tag{9}$$

Here l_i, and $\hat{l}_i$ are actual and predicted ordinal labels respectively.

Results

We conducted a comprehensive set of experiments to evaluate the effectiveness of the ATFES loss function for JSK similarity prediction. Table 1 compares the performance of our proposed model (ATFES) against various baseline methods across different representations, with results measured by MALE and RMSLE. The ATFES loss, using BERT embeddings with all features included, achieves the lowest MALE (0.662) and RMSLE (0.881), demonstrating superior predictive accuracy compared to other baseline methods. Although ORDSIM shows competitive performance, our model with ATFES loss outperforms it by achieving lower error rates on both metrics.

Table 2 presents the confusion matrix for the best-performing configuration of the proposed model, showing the distribution of predictions across all ordinal classes. The diagonal cells, highlighted to indicate True Positives, reflect high prediction accuracy for each class. Off-diagonal cells represent misclassifications, with colors indicating the degree of deviation from the actual class. The confusion matrix reveals that predicted labels maintain ordinal relationships, as, for example, predictions for actual label 4 mostly fall under labels 3 and 4, with a higher number of true positives for label 4. This demonstrates the model's ability to preserve ordinality in predictions.

While both ORDSIM and our proposed model achieve similar test MALE and RMSLE scores, Figure 2 illustrates their performance specifically for ordinal label 4. Since label 4 represents the most critical JSK pairs, accurate performance for these cases is essential. The ORDSIM model tends to centralize predictions, as seen in the left side of the distribution, where labels 3 and 4 have nearly equal numbers of predictions when the true label is 4. This centralization occurs because the loss function does not prioritize label 4's importance. Conversely, the ATFES loss function places greater emphasis on the highest similarity buckets, resulting in a higher count of true positives for label 4. While ordinality is preserved in both cases, the proposed model is better with fewer instances predicted as label 3 compared to the ORDSIM model.

Conclusion and Future Work

We presented a neural network-based ordinal regression model for JSK similarity prediction. Our proposed model outperforms traditional ordinal regression methods by a substantial margin highlighting the benefits of focusing on high-relevance matches. Future work will explore the use of attention mechanisms to further enhance performance and the integration of hierarchical job taxonomies into the embedding process.

Acknowledgments

Dr. Hasan's research is supported by a National Science Foundation (NSF) grant numbered 2417275.

References

Cao, W.; Mirjalili, V.; and Raschka, S. 2020. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*.

Castagnos, F.; Mihelich, M.; and Dognin, C. 2022. A simple log-based loss function for ordinal text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, 4604–4609.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Fuchs, G.; Acriche, Y.; Hasson, I.; and Petrov, P. 2020. Intent-Driven Similarity in E-Commerce Listings.

Kabir, M. A.; Abdelfatah, K.; He, S.; Korayem, M.; and Hasan, M. A. 2024. Forecasting Application Counts in Talent Acquisition Platforms: Harnessing Multimodal Signals using LMs. In *Proceedings of the IEEE International Conference on Big Data*.

Kabir, M. A.; Hasan, M. A.; Mandal, A.; Tunkelang, D.; and Wu, Z. 2022. ORDSIM: Ordinal Regression for E-Commerce QuerySimilarity Prediction. In *Proceedings of the International Workshop on Interactive and Scalable Information Retrieval methods for eCommerce (ISIR-eCom)*.

Mandal, A.; Khan, I. K.; and Kumar, P. S. 2019. Query Rewriting using Automatic Synonym Extraction for E-commerce Search. In *eCOM@SIGIR*.

Meng, Y.; Huang, J.; Wang, G.; Zhang, C.; Zhuang, H.; Kaplan, L.; and Han, J. 2019. Spherical Text Embedding.

Qi, S.; Wu, D.; and Mamoulis, N. 2016. Location Aware Keyword Query Suggestion Based on Document Proximity. *IEEE Transactions on Knowledge and Data Engineering*, 28: 82–97.

Ramalingam, G.; Logeswari, S.; Kumar, M.; Prabakaran, M.; Nishant, N.; and Ahmed, S. A. 2024. Machine learning classifiers to predict the quality of semantic web queries. *The Scientific Temper*, 15(01): 1777–1783.

Rennie, J.; and Srebro, N. 2005. Loss functions for preference levels: Regression with discrete ordered labels.

Shi, X.; Cao, W.; and Raschka, S. 2021. Deep Neural Networks for Rank-Consistent Ordinal Regression Based On Conditional Probabilities. arXiv:2111.08851.

Sun, X.; Meng, Y.; Ao, X.; Wu, F.; Zhang, T.; Li, J.; and Fan, C. 2021. Sentence Similarity Based on Contexts.