Multi-Objective Ranking to Boost Navigational Suggestions in eCommerce AutoComplete

Sonali Singh
ssonl@amazon.com
Amazon

Sachin Farfade
sfarfade@amazon.com
Amazon

Prakash Mandayam Comar
prakasc@amazon.com
Amazon

ABSTRACT

Query AutoComplete (QAC) helps customers complete their search queries quickly by suggesting completed queries. QAC on eCommerce sites usually employ Learning to Rank (LTR) approaches based on customer behaviour signals such as clicks and conversion rates to optimize business metrics. However, they do not exclusively optimize for the quality of suggested queries which results in lack of navigational suggestions like product categories and attributes, e.g., "sports shoes" and "white shoes" for query "shoes". We propose to improve the quality of query suggestions by introducing navigational suggestions without impacting the business metrics. For this purpose, we augment the customer behaviour (CB) based objective with Query-Quality (QQ) objective and assemble them with trainable mixture weights to define multi-objective optimization function. We propose to optimize this multi-objective function by implementing ALMO algorithm to obtain a model robust against any mixture weight. We show that this formulation improves query relevance on an eCommerce QAC dataset by at least 13% over the baseline Deep Pairwise LTR (DeepPLTR) with minimal impact on MRR and results in a lift of 0.26% in GMV in an online A/B test. We also evaluated our approach on public search logs datasets and got improvement in query relevance by using query coherence as QQ objective.

CCS CONCEPTS

• **Computing Methodologies** → **Ranking**; • **Information Systems** → *AutoComplete*.

KEYWORDS

autocomplete, navigational, learning to rank, multi-objective

ACM Reference Format:
Sonali Singh, Sachin Farfade, and Prakash Mandayam Comar. 2023. Multi-Objective Ranking to Boost Navigational Suggestions in eCommerce AutoComplete. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3543873.3584649

1 INTRODUCTION

QAC helps customers articulate their search queries based on the query fragment they typed in the search box. For example, if a customer types "shir" on an eCommerce search, QAC will suggest "shirts, shirts for men, shirts latest, etc.", as possible choices for the customer to select from. QAC enables the customer to type the query in minimum keystrokes reducing query typing by 25% [22]. Historically, QAC has been formulated as a ranking problem on the list of queries typed by customers in the past. This problem is solved typically using a retrieve-and-rank approach, where, given a prefix entered by a customer in the search box, a subset of past queries that match with the prefix are retrieved and further ranked using LTR-based approaches. The ranking strategies optimize on a combination of customer behaviour signals such as clicks, Gross Merchandise Value (GMV), searches, etc. QAC is often a difficult task due to varied search intents and huge query logs to retrieve and rank from.

Many customers are not proficient in using search navigational tools to narrow down the selection based on product category or attribute filtering. QAC could serve as an important tool to help customers navigate to the intended product category and attributes by providing navigational suggestions. Navigational query suggestions educate new customers about the range of selections available to them along with assisting to narrow down the selection based on their needs right at the beginning of the shopping journey. The navigational queries are defined as follows: *These are keyword suggestions that consist of brand name (Nike, Sony, etc.), product name (shirt, watches, etc.), gender, category (running shoe, tennis shoe), and attributes.* These queries help customers with broad searches to funnel down selection based on the product type or attributes like material, style, etc. In fashion shopping journey, these are important parameters in purchase decision. In addition to the navigational needs, typically the search logs contain many poorly formed queries which impact the overall QAC experience. For example, for prefix "sho", search logs contains many noisy completions such as "shoshe man", "shoe mens", etc. We mitigate this by introducing the notion of query coherence where we define *coherence as the probability of the flow of tokens in a query.* QAC serves as an early feature in search and the presence of navigational and coherent suggestions caters to diverse intents of customers and overall provide higher quality suggestions.

We propose to improve the QAC experience by increasing the quality of suggestions in addition to clicks and sales-optimized suggestions. To validate this hypothesis, we propose an ML model that learns to boost the ranking of high-quality query suggestions. We do this by introducing an additional QQ objective that optimizes query relevance and call this modified framework, multi-objective ranking (MOR). We combine the base objectives using mixture weights, thereby converting the problem to scalar function optimization [17]. We first tune the mixture weights treating them as hyper-parameter, however, this tends to over-fit to one of the loss functions [18]. We, therefore, adopt an agnostic approach called ALMO [11] where we jointly learn mixture weights along with the

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9419-2/23/04.
https://doi.org/10.1145/3543873.3584649

model parameters to avoid overfitting on any base objective. MOR is a generic framework and can be applied to any LTR-based QAC approach. We apply the MOR approach on DeepPLTR and show through offline experimentation that it improves the query relevance in the eCommerce dataset by 13% and Mean Reciprocal Rank (MRR) by 2%. We also demonstrate results on AOL public search dataset, where we report a 2.8× improvement on query relevance over the baseline DeepPLTR.

2 RELATED WORK

QAC systems are designed with two broad approaches, (a) query retrieval and ranking (b) fully generative. Retrieval and Ranking based approaches were initially heuristic and based on the Most Popular Completion algorithms [3]. Later, learning-based approaches were used to rank that leveraged time-sensitive signals and user feedback signals [7, 16]. Generative approaches use either heuristics (templates, nn grams, etc.) [4, 24] or generative Neural LM [19, 25] to synthesize full suggestions. More details are in survey [8].

Our work focuses on improving the quality of suggestions in QAC without losing on query clicks and downstream sales in the eCommerce space. [15] studies importance of QAC ranking quality on user interaction patterns. We leverage popular learning to rank objective functions [6, 9, 10] and modify them to boost high-quality suggestions. Research by Jian et al 2010 [5] studies query-dependent loss function where they jointly learn ranking along with query categorization. They study ranking for three query categories -: navigational, informational, and transactional queries. Some groups have studied multi-objective optimization for catering to many notions of query relevance simultaneously. [23] studies combining multiple relevances into single graded relevance. [12] show how the correlations between multiple facets of relevance affect multi-criteria ranking. We first assign a relevance score to each query that reflects how navigational or coherent the query is. We further build an ML-based approach, that uses these scores and learns to rank relevant suggestions higher using multi-objective ranking. We adopt the ALMO algorithm [11] where instead of fixed mixture weights for each ranking objective, the weights are learnable. This optimizes the net loss for any combination of mixture weights, which includes the worst combination of mixture weights and hence is risk-averse.

3 DEEPPLTR MODEL

In this work, we adopt DeepPLTR model [26] as our baseline QAC approach. The model is an NN-based ranker that optimizes customer behaviour using a pairwise loss function that learns to rank an accepted (clicked or fully typed) keyword higher than the rejected (non-clicked) keyword. Within a session, for each keystroke, the accepted keyword (labeled as positive ✓) is sampled and paired with all the rejected keywords (labeled as negative X). The positive and the negative keywords are featurized and given as input to a Siamese Neural Network with a pairwise cross-entropy loss function as the head. Each keyword pair in a session is assigned a weight as per the optimization criterion. To optimize for customer behaviour, it is set as a number proportional to GMV generated from clicking on the positive keyword of the pair.

For a given prefix p, let k denote a keyword that completes p and $\mathbf{x}_k^p \in \mathcal{R}^d$ denote feature vector that summarizes the relationship between p and k. Let $f : \mathcal{R}^d \rightarrow \mathcal{R}$ denote an acceptance function that maps x_k^p to a score that reflects the user's affinity of accepting the recommended keyword k towards completing the prefix p, such that it results in conversion. Let k_+ and k_- denote positive and negatively labeled keywords that complete the prefix p and S denote all such keyword pairs. To learn f, we optimize CB objective using a pairwise loss function $L_{CB} : \mathcal{R}^d \times \mathcal{R}^d \rightarrow \mathcal{R}$ that takes as input a keyword pair (k_+,k_-) such that L_{CB} is minimized if f assigns a higher score to k_+ compared to k_-. Let x_{k+}^p and x_{k-}^p denote the features of prefix keyword tuple (p,k_+) and (p,k_-) respectively. r_{k+} and r_{k-} denote the rank or position of k_+ and k_- respectively shown to the customer in the past sessions. An acceptance function f is constructed such that $f_{k+} = f(x_{k+}^p) > f(x_{k-}^p) = f_{k-}$ by minimizing following pairwise loss:

$$L_{CB} = \frac{1}{|S|} \sum_{(k+,k-)\in S} w_{k+} \left[log(1 + e^{(f_{k-} - f_{k+})}) * |\Delta_n| \right] \quad (1)$$

where $\Delta_n = \frac{1}{log(1+r_{k+})} - \frac{1}{log(1+r_{k-})}$ denote the difference in reciprocal rank of the positive and negative keywords. The term $|\Delta_n|$ amplifies loss for keyword pairs separated by a higher margin in the ranked list, driving to learn on highly relevant keywords ranked very low in the list. The weight w_{k+} ensures that the pairwise loss for sessions where the selected keyword resulted in a non-negative purchase (GMV) is as low as possible.

4 MULTI-OBJECTIVE AUTOCOMPLETE

We introduce Multi-Objective Ranking (MOR) for QAC that leverages all the advantages of existing LTR- based approaches and, in addition, improves the quality of query suggestions.

4.1 Multi-Objective Ranking

In MOR, we define a query relevance score which is further used in a modified LTR objective function explained in subsection 4.1.2 designed to uprank suggestions with a higher query relevance score while improving on click rate and GMV. Details of query relevance score and modified objective function are:

4.1.1 Query Relevance Score: We assign a prefix-independent, static relevance score (denoted as y_k^{rel}) to each query, similar to the notion of query-independent document quality score [20]. This score measures how relevant the queries are for the QAC task. For eCommerce search, we define relevance by measuring how navigational the query is. While for public search logs, relevance is measured by the coherence of the sequence of tokens in the query.

(1) Navigational Score: In order to improve query quality, we recommend queries with more navigational terms without losing on click and conversion rate. We achieve this by modifying the objective function described in Section 3 such that the acceptance function f is high for queries with more navigational terms. For a keyword k, the navigational score reflects the extent to which navigational terms are present in the keyword. This score is independent of the prefix p

and only depends on the terms/words in the keyword k. We use query understanding tools to identify navigational words/tokens like brand name, product, gender, category, or attributes. Navigational score for the query is then calculated based on navigational terms in each keyword and the keyword's popularity.

(2) Query Coherence Score We also extend MOR to public search logs where we measure relevance in terms of coherence of suggestions in top slots. We define query coherence score as how probable the sequence of words in a query is or how well-formed the query is. As an example, consider two query completions q1:"tripadvisor.com" and q2:"triple a" for prefix "trip". q1 which is a url is more coherent as compared to q2 and hence will be assigned a higher coherence score. We use a large pre-trained language model (GPT-2) [1] to compute the coherence score. It is calculated as inverse of perplexity [2] for a query.

4.1.2 Multi-Objective pairwise loss. We propose a multi-objective pairwise loss function that jointly models the customer behaviour (GMV, clicks) and the query relevance score of the recommended keywords. We retain the CB objective as in eq. 1 and add an extra QQ objective that boosts the rank for high-quality suggestions. Keeping the two losses separate and combining them by weights has the following benefits a) easy to implement and measure the trade-off varying one loss over the other b) retaining all the advantages of the CB objective.

Joint Regression Learning: One straight-forward approach to jointly optimize CB and QQ objective is to directly minimize the mean squared error (MSE) between the predicted acceptance scores f and fixed query relevance score y_k^{rel} for each pair as given below:

$$L_{reg} = \frac{1}{|S|}\sum_{(k+,k-)\in S}\Bigg[\lambda_1\left\{w_{k+} * log(1+e^{(f_{k-}-f_{k+})})|\Delta_n|\right\} + \lambda_2\left\{f_{k+} - y_{k+}^{rel}\right\}^2\Bigg] \quad (2)$$

We consider only the positive keyword while computing MSE because we want the positive keyword's predicted score f_{k+} to be closest to its relevance score y_{k+}^{rel} and do not want to do such explicit minimization for negative keyword. The weights λ_1 and λ_2 can either be tuned as a hyper-parameter or can be learned using ALMO. L_{reg} is computed over a batch of random samples S. Although the base losses can be optimized individually without any issues for their respective task, the combination is tricky to optimize. Particularly, we found it difficult to tune the λ_2 for the above loss as the MSE term has both high variance and magnitude as compared to CB-based pairwise loss. The pairwise loss stays in the range of [0,1] due to two factors a) The cross-entropy loss stays between [0,1] as the probability of $f_{k+} > f_{k-}$ exceeds 0.5 in most cases b) The multiplier Δ_n stays in [0,1] due to the difference between the reciprocal of rank. The net effect results in averaged pairwise loss for a batch in between [0,1]. This is not the case with MSE loss where predicted score f_{k+} and y_{k+}^{rel} varies significantly from sample to sample which makes it harder to finetune λ_2. Moreover, ALMO requires all base losses to be bounded. Due to these disadvantages, we introduce correlation loss replacing the regression (MSE) loss for the QQ objective.

Table 1: Percentage change in MRR and correlation metric of moDPLTR over DeepPLTR Baseline on eCommerce dataset.

Model	MRR	Corr(f,y_k^{rel})
moDPLTR		
Reg	-6.22%	+43.42%
Corr	-4.14%	+32.45%
moDPLTR ALMO		
Reg	+3.31%	+4.82%
Corr	+2.48%	+13.15%

Joint Correlation Learning: Instead of direct minimization of mean square error between f_{k+} and y_{k+}^{rel} for each sample, we maximize the linear correlation between the acceptance function ($\mathbf{f}_{k+}$) and the query relevance scores ($\mathbf{y}_{k+}^{rel}$) in a batch. Note that $\mathbf{f}$ and $\mathbf{y}$ are vectors of predicted and navigational scores for positive keywords in a batch. As we need to minimize the net loss, so we add the negative of this correlation in the existing CB objective. This choice for the QQ objective, not only bounds the correlation loss between [-1,1] but also minimizes the variance in loss due to the correlation computed over a batch (vs. regression loss computed for each pair). Let $Corr(x, y)$ denote the correlation between x and y, and S be training samples in a batch then,

$$L_{corr} = \lambda_1\left\{\frac{1}{|S|}\sum_{(k+,k-)\in S} w_{k+} * log(1+e^{(f_{k-}-f_{k+})})\ |\Delta_n|\right\} - \lambda_2\left\{Corr(\mathbf{f}_{k+}, \mathbf{y}_{k+}^{rel})\right\} \quad (3)$$

Optimizing multi-objective pairwise loss as stated above involves searching λ over a range [0,1] and finally training with fixed λ that gives minimum validation error. This way we may get a solution that minimizes the ensemble of losses, but may significantly hurt a loss at the cost of over-optimizing on other losses. The learners' preference towards an objective can vary over time due to multiple factors which are not taken care of by fixed weights. Moreover, its expensive computationally to obtain a Pareto-optimal solution with multiple base objectives [13, 14]. We, therefore, adopt the ALMO [11] algorithm that proposes to train a model, robust against any combination of mixture weights by converting the problem to min-max optimization. It also guarantees the solution to be Pareto-optimal.

5 EXPERIMENTAL EVALUATION

We present both offline and online evaluations of Multi-Objective Ranking framework in this section. We apply the MOR framework on DeepPLTR model and call the resultant model as Multi-Objective DeepPLTR (or moDPLTR).

5.1 Offline evaluation

5.1.1 Training and Evaluation Samples.

(1) eCommerce Data: We use user-anonymized internal search logs from an eCommerce site for testing MOR framework, both in an offline and online setting. Customer behaviour signals such as aggregated clicks, sales, and contextual signals (based on the user's environment) are used as features and navigational scores are employed for query relevance in the QQ objective. As described in section 3, we sample positive and negative pairs from search logs and generated 12M samples for training and 2M for validation aggregated over 2 days. For evaluation, each candidate is scored in the 100 most popular candidates list per prefix and finally the top 10 candidates are selected. Finally, we evaluated on 50k prefixes sampled from a day.

(2) AOL Data: To study the impact of MOR on web search queries, we experiment with publicly accessible AOL search logs [21]. Similar to [19], training and test data are split on timestamps. Further, we uniformly sample prefixes from these queries and build a mapping of prefixes to the most popular query candidate list. For each query in logs, the searched query is treated as a positive keyword and paired with other queries (treated as negative keywords) in the most popular candidate list. Query coherence scores are used as relevance scores in the QQ objective. 15M data-points are used for training and 90k prefixes for evaluation. As features, we use sentence embedding from pre-trained language models, aggregated clicks, and cosine similarity of a query with immediate past query in the same session.

5.1.2 Model Architecture. The architecture of moDPLTR is similar to DeepPLTR i.e. RankNet-based fully connected Neural Network with a change in ranking head to accommodate QQ objective. The detailed architecture and layer size is in fig. 2. For both the positive and negative sample pair, we see a siamese NN architecture with weight sharing. Similar to DeepPLTR, we leverage the contextual layer for contextual features. The ranking scores from the positive and negative dense layers are given as input to the pairwise loss. In addition, the ranking score of the positive keyword is also given as input to QQ objective function for computing correlation or regression loss. Finally, the pairwise loss for the CB objective and correlation/regression loss for the QQ objective are assembled with mixture weights.

5.1.3 Offline Metrics and Results. Tables 1 and 2 compares the performance of moDPLTR against the DeepPLTR model (baseline) on eCommerce and AOL datasets respectively. For the multi-objective optimization, the mixture weights are treated as hyper-parameter in moDPLTR and are learned using the ALMO algorithm in the moDPLTR ALMO model. We choose two primary metrics for evaluation- MRR for tracking performance w.r.t CB objective and correlation metric denoted as Corr(f,y_k^{rel}) for measuring query relevance for QQ objective. In addition, we also use some secondary metrics such as text and category diversity for measuring diversity of QAC suggestions. Details of metrics used for the evaluation are:

Prefix ="shir" from eCommerce dataset trained for upranking suggestions with high navigational scores		Prefix = "nortan ant" from AOL dataset trained for upranking suggestions with high query coherence score	
DeepPLTR	**moDPLTR ALMO (corr)**	**DeepPLTR**	**moDPLTR ALMO (corr)**
shirts for men	shirts for men	norton antivirus	norton antivirus
shirts latest design	shirts casual	norton anti virus	norton antivirus support
shirts combo pack	shirts for women	norton anti-virus	norton antivirus firewall protection
shirts for women	shirts full-sleeves	norton antivirus update	norton antivirus protection
shirt for man	shirts latest	norton anti virus.com	norton antivirus 2006

Figure 1: Experience comparing DeepPLTR suggestions with moDPLTR ALMO suggestions on eCommerce data on the left and AOL data on the right.

Table 2: Results comparing the performance of moDPLTR with DeepPLTR on AOL data.

Model	MRR	Corr(f,y_k^{rel})
DeepPLTR	0.485	0.077
moDPLTR		
Corr	0.479 (-1.23%)	0.095 (+23.37%)
moDPLTR ALMO		
Corr	0.438 (-9.6%)	0.222 (+188.3%)

MOR metric definitions The metrics used for offline evaluation are defined as follows:-

(1) MRR: Standard Mean Reciprocal Rank on sessions on all prefix p, i.e., $\frac{\sum_{\forall p} 1*RR}{\sum_{\forall p} 1}$ where RR is reciprocal of the rank of selected query in the ranked list of suggestions. RR is 0 if selected query is not present in the ranked list.
(2) Correlation Metric: Corr(f,y_k^{rel}) denotes mean correlation over all prefixes between acceptance scores f and query relevance scores y_k^{rel} for top 10 QAC suggestions.
(3) Text Diversity:It measures the fraction of unique tokens in the top 10 suggestions averaged over total number of tokens. The diversity computed for each suggestion is weighted by its reciprocal rank (RR) to attribute maximum diversity to top slot suggestion.
(4) Category Div: Category Diversity measures the number of unique categories in the top 10 suggestions per prefix.

eCommerce data: For both the versions of moDPLTR model (without ALMO), we see a significant increase in correlation metric (+38%) with a drop in MRR (-5%) on average. This trade-off between MRR and correlation metric is a common challenge in multi-objective optimization where fixed weights tend to hurt an objective function while over-optimizing on other components. We mitigate this by adopting the ALMO algorithm to learn the mixture

weights. Both versions of moDPLTR ALMO models improve both on MRR and correlation metric as compared to baseline. Correlation objective performs better on an average than regression due to challenges with the latter as explained in Section 4.1.2. We also show results on diversity based metric in table 3. We observe a lift in both text and category based diversity for all model variations as compared to baseline.

AOL Data: We report performance on only the correlation loss as the regression loss for the QQ objective is unsuitable given that query coherence score is a probability while the acceptance score is real-valued number. We see a marginal decline in MRR both for moDPLTR (-1%) and moDPLTR ALMO (-9%), however, the ALMO variation gives a major improvement in correlation metric(+188%) vs the fixed weight moDPLTR (+23%).
moDPLTR ALMO (Corr) is chosen as the best-performing model as it gives the highest improvement in correlation metric with comparable performance in MRR as compared to the other model variations. In fig. 1, we compare the QAC performance of DeepPLTR with moDPLTR ALMO (Corr) on two popular prefixes p_1="shir" and p_2="nortan ant". For prefix p_1, we see more attributes such as 'casual' (shirt category), and 'full-sleeves' (shirt feature) in suggestions from moDPLTR as compared to the baseline. This shows that correlating acceptance scores with navigational scores helps in upranking queries with more navigational scores higher. Similarly, for prefix p_2 from AOL search logs, we correlate acceptance scores with query coherence scores. We observe that baseline shows some misspelled variations of 'antivirus' such as 'anti virus', 'anti-virus' which are replaced by more coherent queries in the moDPLTR model.

5.2 Online Evaluation (A/B test)

We evaluated the moDPLTR ALMO model with joint correlation learning on the search bar of a large eCommerce platform search as part of online A/B test. The model was tested for 4 weeks on 1M prefixes, which resulted statistically significant lift of +0.26% in GMV and a +0.41% lift in shopping items' diversity.

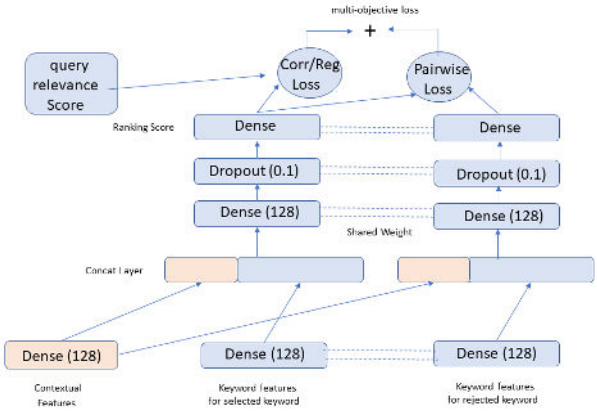

Figure 2: Model architecture for Multi-Objective Ranking.

Table 3: Percentage change in text and category diversity of moDPLTR over DeepPLTR Baseline on eCommerce dataset.

Model	Txt Div.	Cat Div.
moDPLTR		
Reg	+2.5%	+6.98%
Corr	+2.85%	+8.01%
moDPLTR ALMO		
Reg	+1.2%	+0.18%
Corr	+1.5%	+2.70%

6 CONCLUSION

We introduce MOR framework that improves the QAC experience by recommending high-quality suggestions. The proposed ML approach jointly optimizes query relevance and customer-behaviour using a multi-objective criterion. For multi-objective training, instead of selecting fixed mixture weights, we proposed to use ALMO to achieve robustness against any mixture weights. We defined navigational score and query coherence to measure query quality and showed improvement in query relevance with our approach on an eCommerce QAC and public search logs datasets with minimal impact on MRR. Moreover, our approach can we used with any other LTR model for multi-objective ranking .

REFERENCES

[1] 2022. GPT-2 in huggingface. https://huggingface.co/docs/transformers/model_doc/gpt2.
[2] 2022. Perplexity Metric in huggingface. https://huggingface.co/spaces/evaluate-metric/perplexity.
[3] Ziv Bar-Yossef and Naama Kraus. 2011. Context-sensitive query auto-completion. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*. 107–116.
[4] Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. 2011. Query suggestions in the absence of query logs. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 795–804.
[5] Jiang Bian, Tie-Yan Liu, Tao Qin, and Hongyuan Zha. 2010. Ranking with query-dependent loss for web search. In *Proceedings of the third ACM international conference on Web search and data mining*. 141–150.
[6] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.
[7] Fei Cai and Maarten de Rijke. 2016. Learning from homologous queries and semantically related terms for query auto completion. *Information Processing & Management* 52, 4 (2016), 628–643.
[8] Fei Cai, Maarten De Rijke, et al. 2016. A survey of query auto completion in information retrieval. *Foundations and Trends® in Information Retrieval* 10, 4 (2016), 273–363.
[9] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*. 129–136.
[10] Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhi-Ming Ma, and Hang Li. 2009. Ranking measures and loss functions in learning to rank. *Advances in Neural Information Processing Systems* 22 (2009).
[11] Corinna Cortes, Mehryar Mohri, Javier Gonzalvo, and Dmitry Storcheus. 2020. Agnostic learning with multiple objectives. *Advances in Neural Information Processing Systems* 33 (2020), 20485–20495.
[12] Na Dai, Milad Shokouhi, and Brian D Davison. 2011. Multi-objective optimization in learning to rank. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 1241–1242.
[13] Kevin Duh, Katsuhito Sudoh, Xianchao Wu, Hajime Tsukada, and Masaaki Nagata. 2012. Learning to translate with multiple objectives. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1–10.
[14] Parke Godfrey, Ryan Shipley, and Jarek Gryz. 2007. Algorithms and analyses for maximal vector computation. *The VLDB Journal* 16, 1 (2007), 5–28.

[15] Kajta Hofmann, Bhaskar Mitra, Filip Radlinski, and Milad Shokouhi. 2014. An eye-tracking study of user interactions with query auto completion. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 549–558.

[16] Jyun-Yu Jiang, Yen-Yu Ke, Pao-Yu Chien, and Pu-Jen Cheng. 2014. Learning user reformulation behavior for query auto-completion. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 445–454.

[17] Yaochu Jin. 2006. *Multi-objective machine learning*. Vol. 16. Springer Science & Business Media.

[18] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7482–7491.

[19] Gyuwan Kim. 2019. Subword language model for query auto-completion. *arXiv preprint arXiv:1909.00599* (2019).

[20] Christopher D Manning. 2008. *Introduction to information retrieval*. Syngress Publishing,.

[21] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *Proceedings of the 1st international conference on Scalable information systems*. 1–es.

[22] Danny Sullivan. [n. d.]. "How Google autocomplete works in Search". "https://www.blog.google/products/search/how-google-autocomplete-works-search/".

[23] Krysta M Svore, Maksims N Volkovs, and Christopher JC Burges. 2011. Learning to rank with multiple objective functions. In *Proceedings of the 20th international conference on World wide web*. 367–376.

[24] Idan Szpektor, Aristides Gionis, and Yoelle Maarek. 2011. Improving recommendation for long-tail queries via templates. In *Proceedings of the 20th international conference on World wide web*. 47–56.

[25] Sida Wang, Weiwei Guo, Huiji Gao, and Bo Long. 2020. Efficient neural query auto completion. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2797–2804.

[26] Kai Yuan and Da Kuang. 2021. Deep Pairwise Learning To Rank For Search Autocomplete. *arXiv preprint arXiv:2108.04976* (2021).