Medium: Learn to Ranking - A complete guide to ranking using machine learning

Ranking models typically work by predicting a relevance score $s = f(x)$ for input $x = (q, d)$, where q is a query and d is a document.

$$x = (q, d) \rightarrow \boxed{\text{Scoring Model}} \rightarrow s = f(x)$$

Approaches

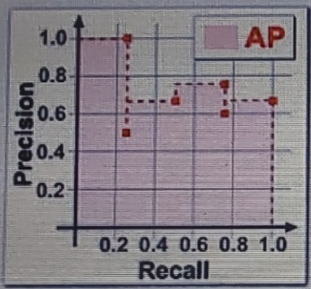
- Vector Space Model: Use embedding (Tf-Idf or BERT) for each document, and compute the relevance score as the cosine similarity.
- Learning to Ranking: Score model is a machine learning model that learns to predict a score s given an input x during a traing phase where some sort of ranking loss is minimzed

Ranking Evaluation Metrics

These metrics are computed on the predict documents ranking, i.e. the k-th top retrieved document is the k-th document with highest predict score s.

Mean Average Precision (MAP)

Binary relevance, i.e where the true score y of the document d can be 0 or 1.



$$AP = \sum_{k=1}^{n} (R_k - R_{k-1}) P_k$$

	1st	2nd	3rd	4th	5th	6th
y_k	1 ✓	0 ✗	1 ✓	1 ✓	0 ✗	1 ✓
P_k	1.00	0.50	0.67	0.75	0.60	0.67
R_k	0.25	0.25	0.50	0.75	0.75	1.00

AP= 0.77

Gived q and $D = \{d_1, \dots, d_n\}$, we check how many of the top k retrived document are relevant ($y=1$) or not ($y=0$), in order to compute precision P_K and recall R_K. For $k = 1 \dots n$ we get different P_K and R_K values that define the precision-recall curve: the area under this curve is the avarege precision (AP)

Finally, by computing the avarage of AP values for a set of m queries, we obtain the mean avarage precision (MAP)

Discounted Comulative Gain (DCG)

Used for task with graded relevance, when the true score y of a document d is a discrete value in a scale measuring the relevance w.r.t a query q. A typical scale is 0 (bad), 1 (fair), 2 (good), 3 (excellent), 4 (perfect).

$$G_k = 2^{y_k} - 1$$

$$D_k = \frac{1}{\log_2(k+1)}$$

$$DCG = \sum_{k=1}^{n} G_k D_k$$

	1st	2nd	3rd	4th	5th	6th
y_k	3	4	1	2	0	1
G_k	7.00	15.00	1.00	3.00	0.00	1.00
D_k	1.00	0.63	0.50	0.43	0.39	0.36

DCG 18.61
NDCG 0.86

Gived q and $D = \{d_1, \dots, d_n\}$, we consider the k-th top retrived document. The gain $G_k = 2^{y_k} - 1$ mensures how useful is this document, while the discount $D_k = \frac{1}{\lg(k+1)}$ penalize documents that are retrieved with lower rank.

The sum of the descount gain terms $G_k D_k$ for $k = 1 \dots n$ is the Discounted Cumulative gain (DCG). To make sure that this score is bound between 0 and 1 we generate the Normalized discounted Cumulative gain (NDCG).

Finally, as for MAP, we usually compute te average DCG or NDCG velues for a set of m queries to obtain a meen value.

Machine learning models for Learning to rank

We need define input, output and loss funtion.

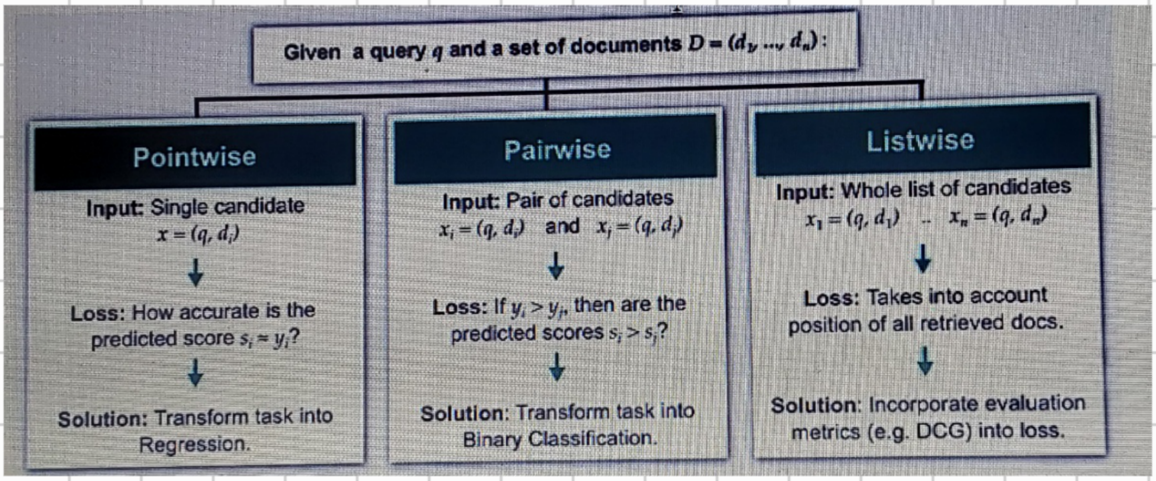
$\hookrightarrow x = (q, d)$ $\hookrightarrow s = f(x)$ $\hookrightarrow$ Pointwise, pairwise and listwise

The choice of the loss funtion is the distinctive element for learning to renk models.

1. **Pointwise**: The total loss is computed as the sum of loss terms defined on each document d_i as the distance between the predict score S_i and the ground truth y_i for $i = 1 \dots n$. By doing this, we trensform our task into a regression problem, where we train a model to predict y.

2. Pairwise: The total loss is computed as the sum of loss terms defined on each pair of documents $d_i\ d_j$, for $i, j = 1 \dots n$. The objective on which the model is trained is to predict whether $y_i > y_j$ or not, i.e which of two documents is more relevant. By doing this, we transform our task into a binary classification problem.

3. Listwise: The loss is directly computed on the whole list of documents with corresponding predicted ranks. In this way, ranking metrics can be more directly incorporated into the loss.



- Basicamente regressão

- mesma importancia para todos os documentos.
- A maioria das vezes temos apenas infos parciais.

- Direta, maximiza avaliação
- Classificação não é diferenciavel