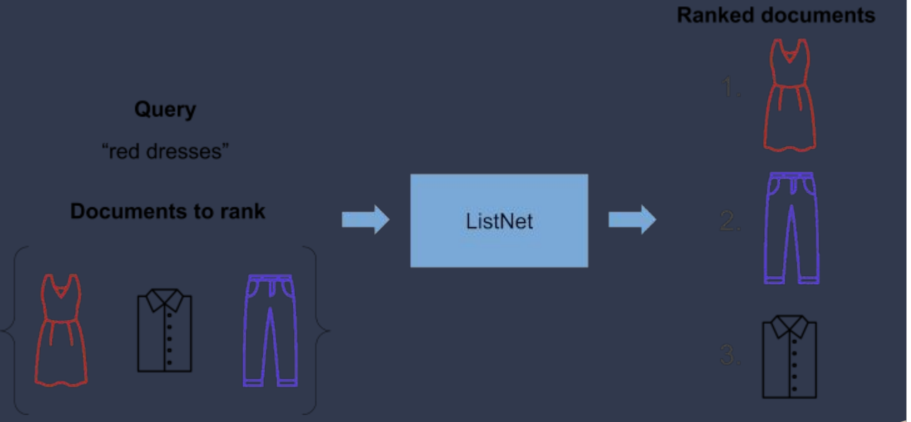
Learning to Rank

Introduction



What is Learning to Rank ?

→ Learning to Rank (LTR) is a class of techniques that apply supervised machine learning (ML) to solve **ranking problems**.

Main difference between LTR and supervised ML :

- Traditional ML solves a prediction problem (classification or regression) on a single instance at a time.
- LTR solves a ranking problem on a list of items. The aim of LTR is to come up with optimal ordering of those items. As such, LTR doesn't care much about the exact score that each item gets, but cares more about the relative ordering among all the items.

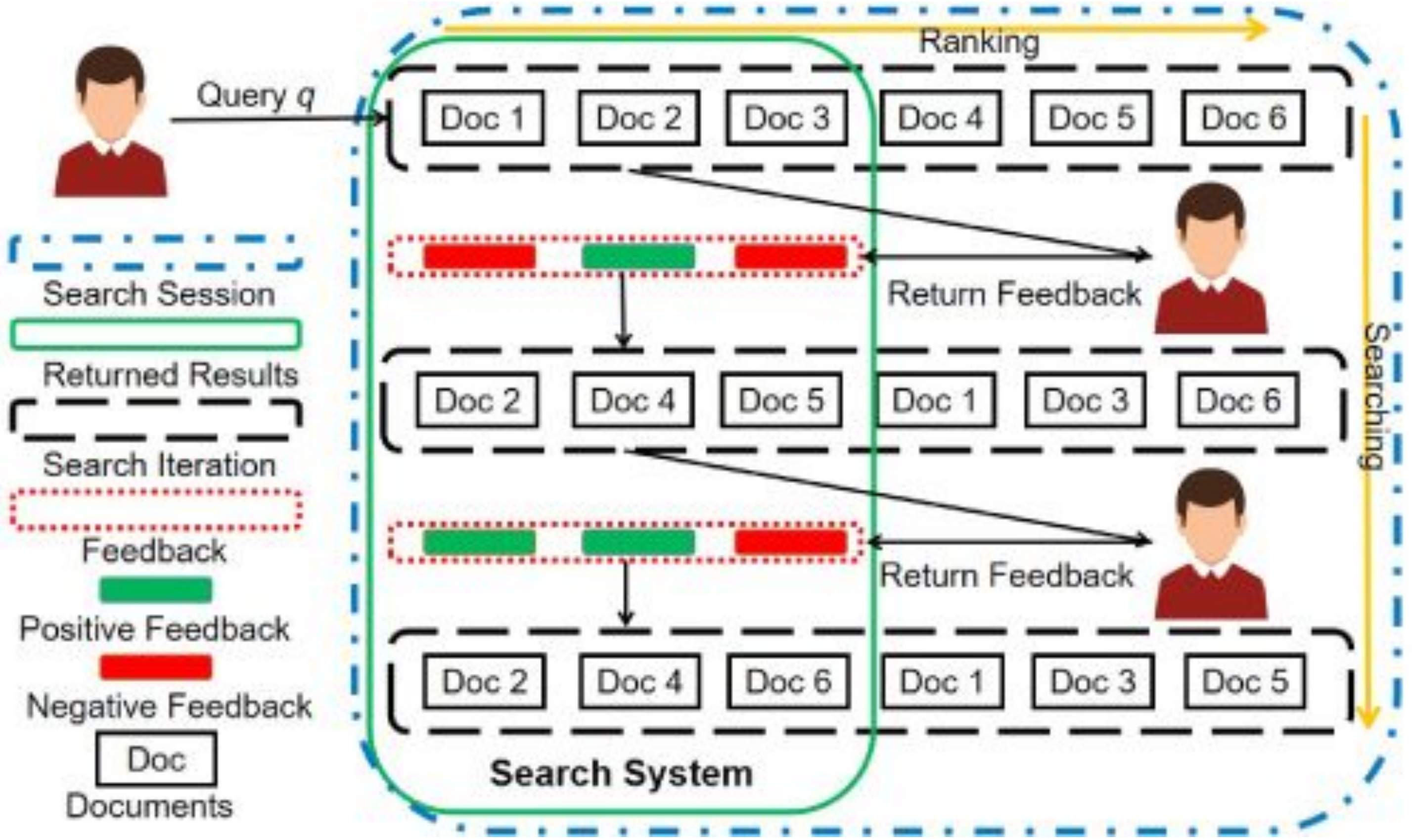
Advantages & Inconvenients

Advantages :

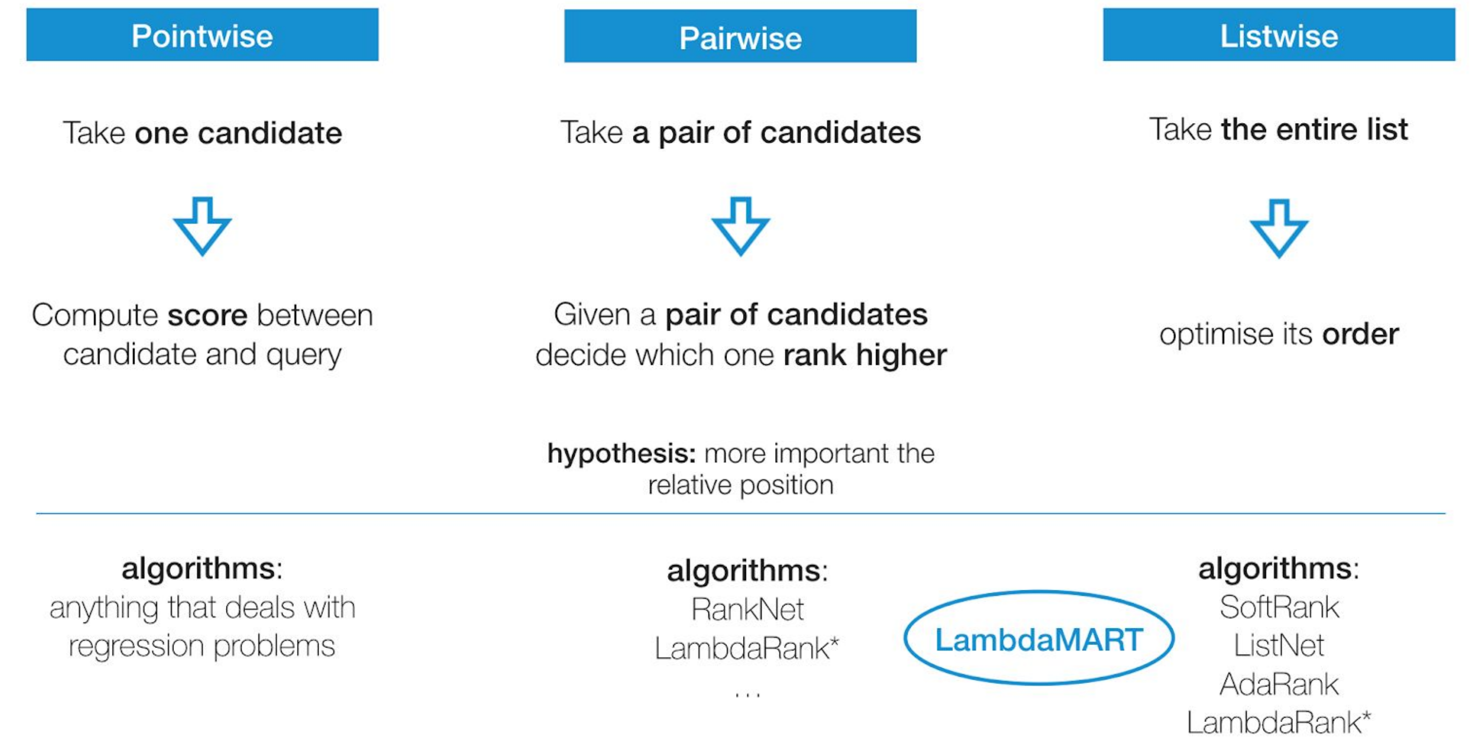
1. Significant increase in search sessions per user
2. Increase in clicks per search
3. Reduction in searches per session

Challenges :

- Defining good features
- Converting search catalog data into effective training sets
- Obtaining relevance judgments (including both explicit judgments by humans and implicit judgments based on search logs)
- Deciding which objective function to optimize for specific applications.
- ...


Query q
Ranking
Doc 1
Doc 2
Doc 3
Doc 4
Doc 5
Doc 6
Return Feedback
Doc 2
Doc 4
Doc 5
Doc 1
Doc 3
Doc 6
Return Feedback
Doc 2
Doc 4
Doc 6
Doc 1
Doc 3
Doc 5
Searching
Search System
Search Session
Returned Results
Search Iteration
Feedback
🟩 Positive Feedback
🟥 Negative Feedback
Doc
Documents

3 LTR Approaches



Pointwise

Take **one candidate**

Compute **score** between candidate and query

Pairwise

Take **a pair of candidates**

Given a **pair of candidates** decide which one **rank higher**

hypothesis: more important the relative position

Listwise

Take **the entire list**

optimise its **order**

algorithms:
anything that deals with regression problems

algorithms:
RankNet
LambdaRank*
...

LambdaMART

algorithms:
SoftRank
ListNet
AdaRank
LambdaRank*

Practical Challenges in Implementing Learning to Rank

You need to decide on the approach you want to take before you begin building your models.

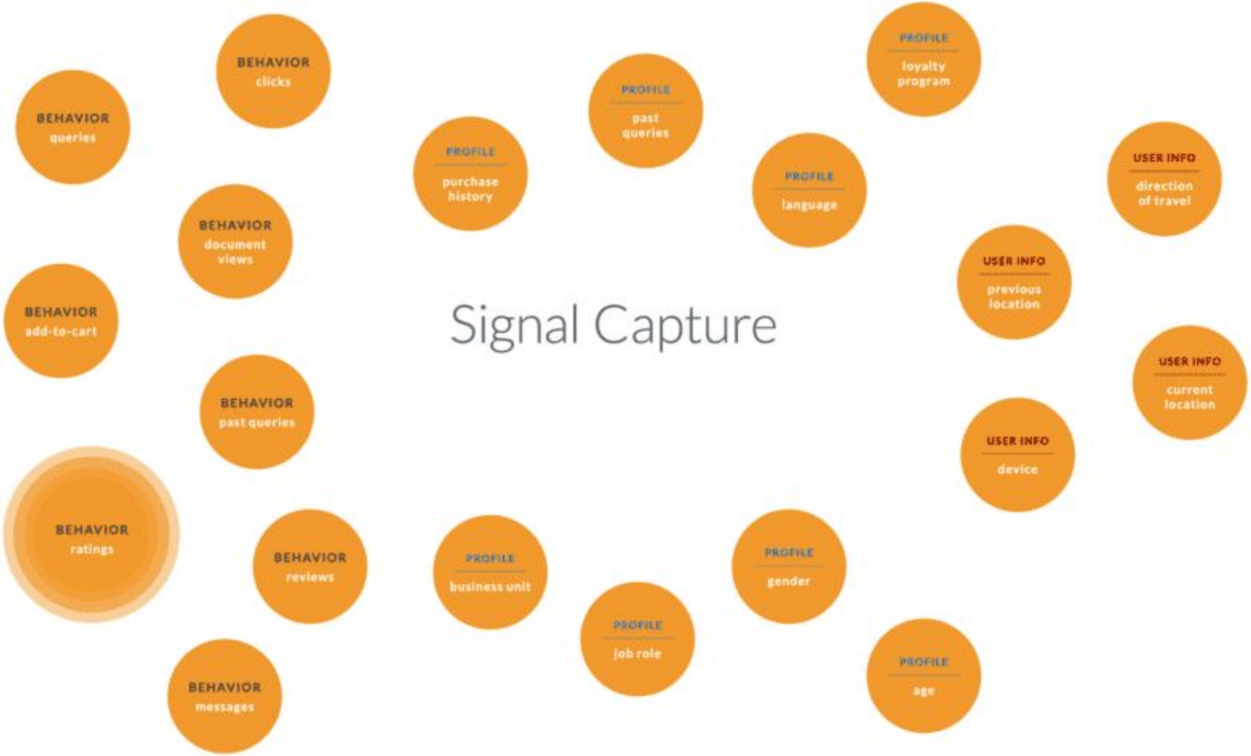
You also need to:

- Decide on the features you want to represent and choose reliable relevance judgments before creating your training dataset.
- Choose the model to use and the objective to be optimized.

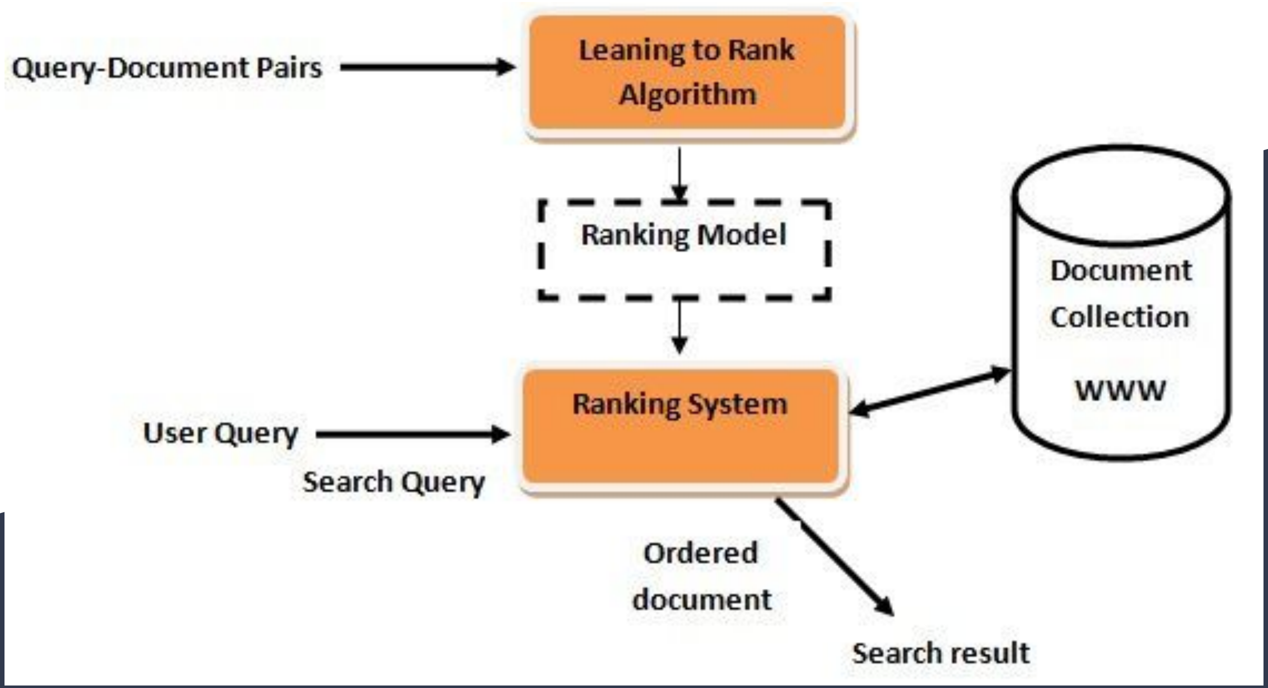
In particular, the trained models should be able to generalize to:

- Previously unseen queries not in the training set
- Previously unseen documents to be ranked for queries seen in the training set.

SIGNAL CAPTURE



Learning to Rank Architecture


Query-Document Pairs
Leaning to Rank Algorithm
Ranking Model
Document Collection
WWW
Ranking System
User Query
Search Query
Ordered document
Search result