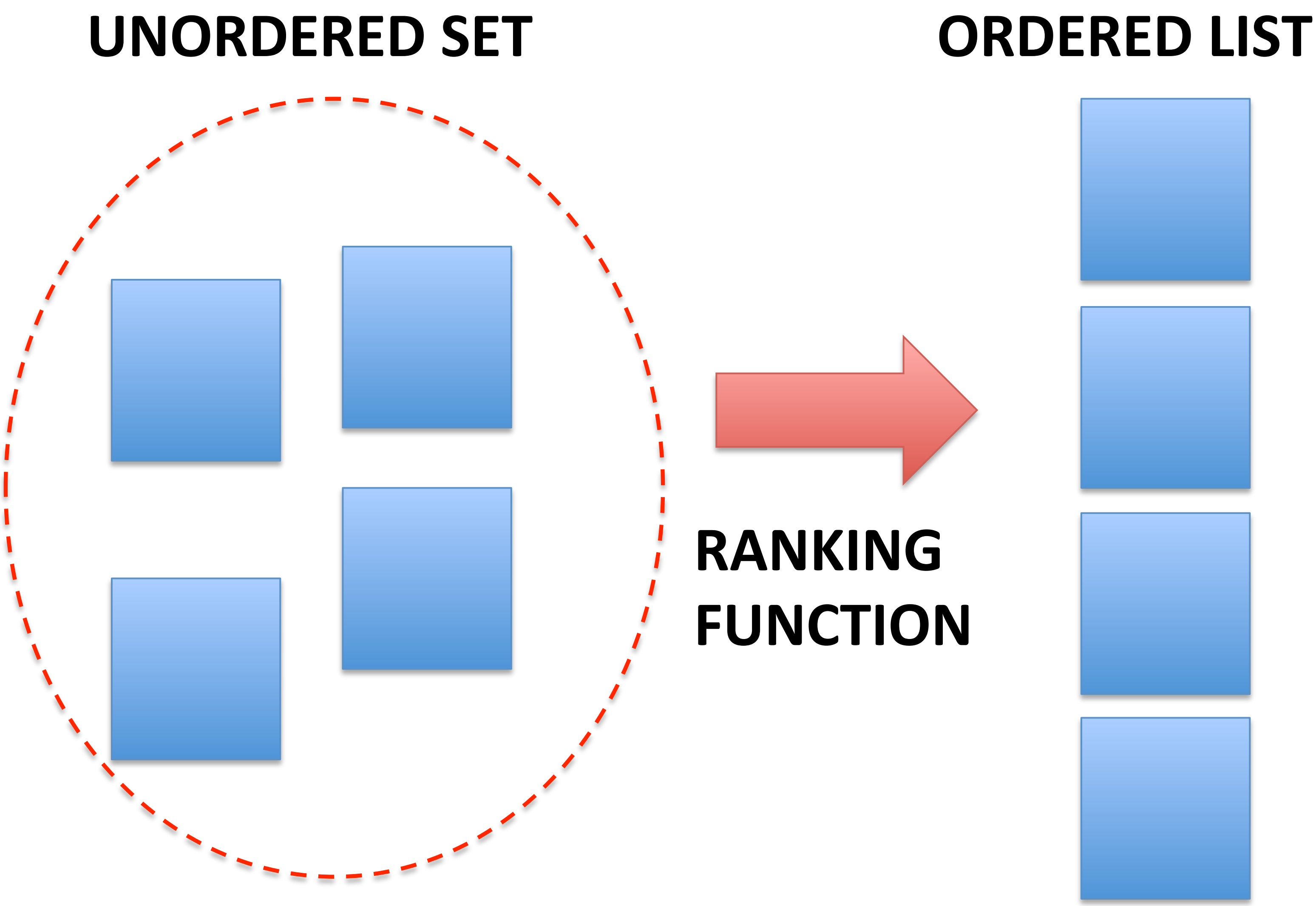
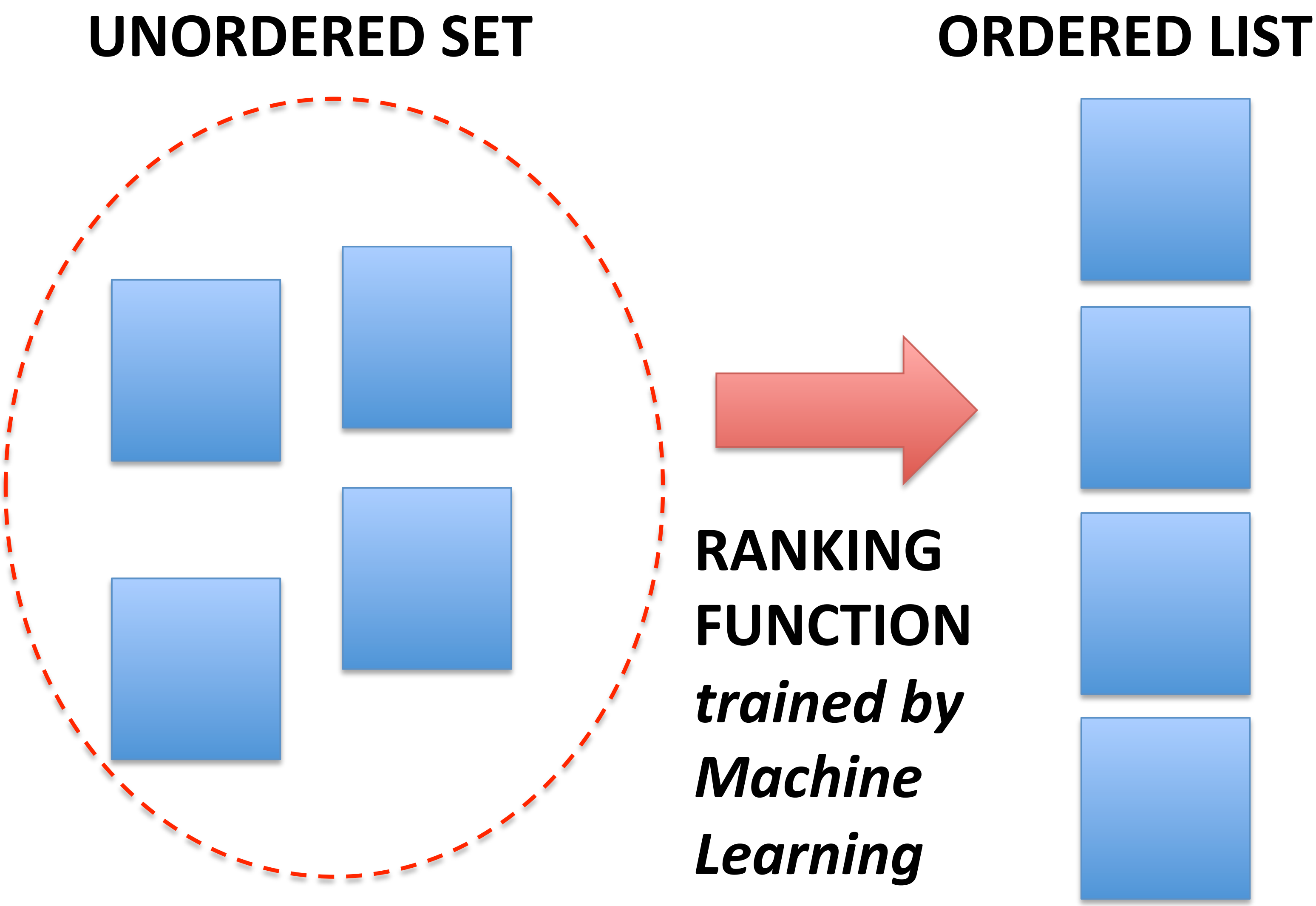
# Learning to Rank

Kevin Duh
Johns Hopkins University
June 2019

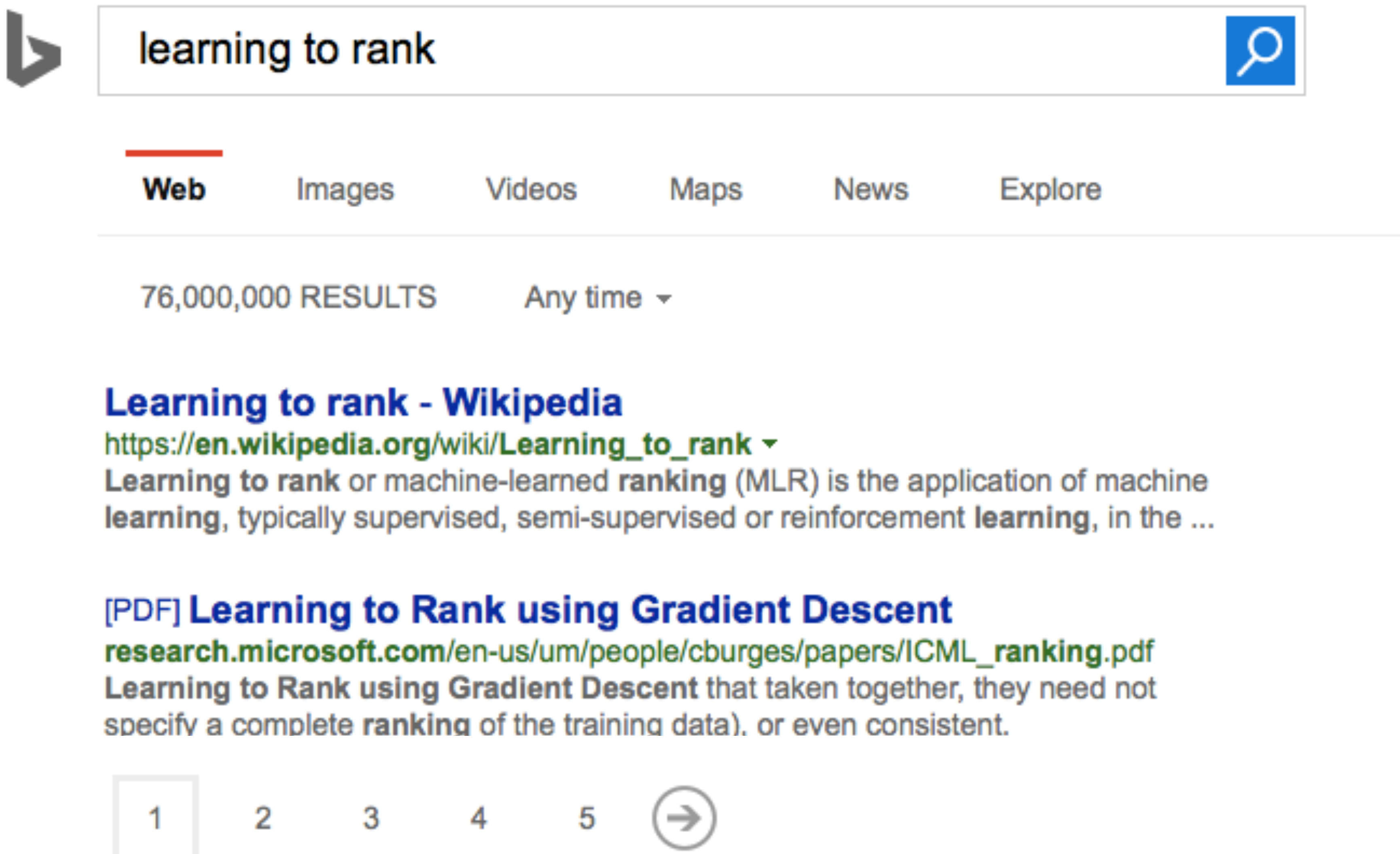
# What is Ranking?



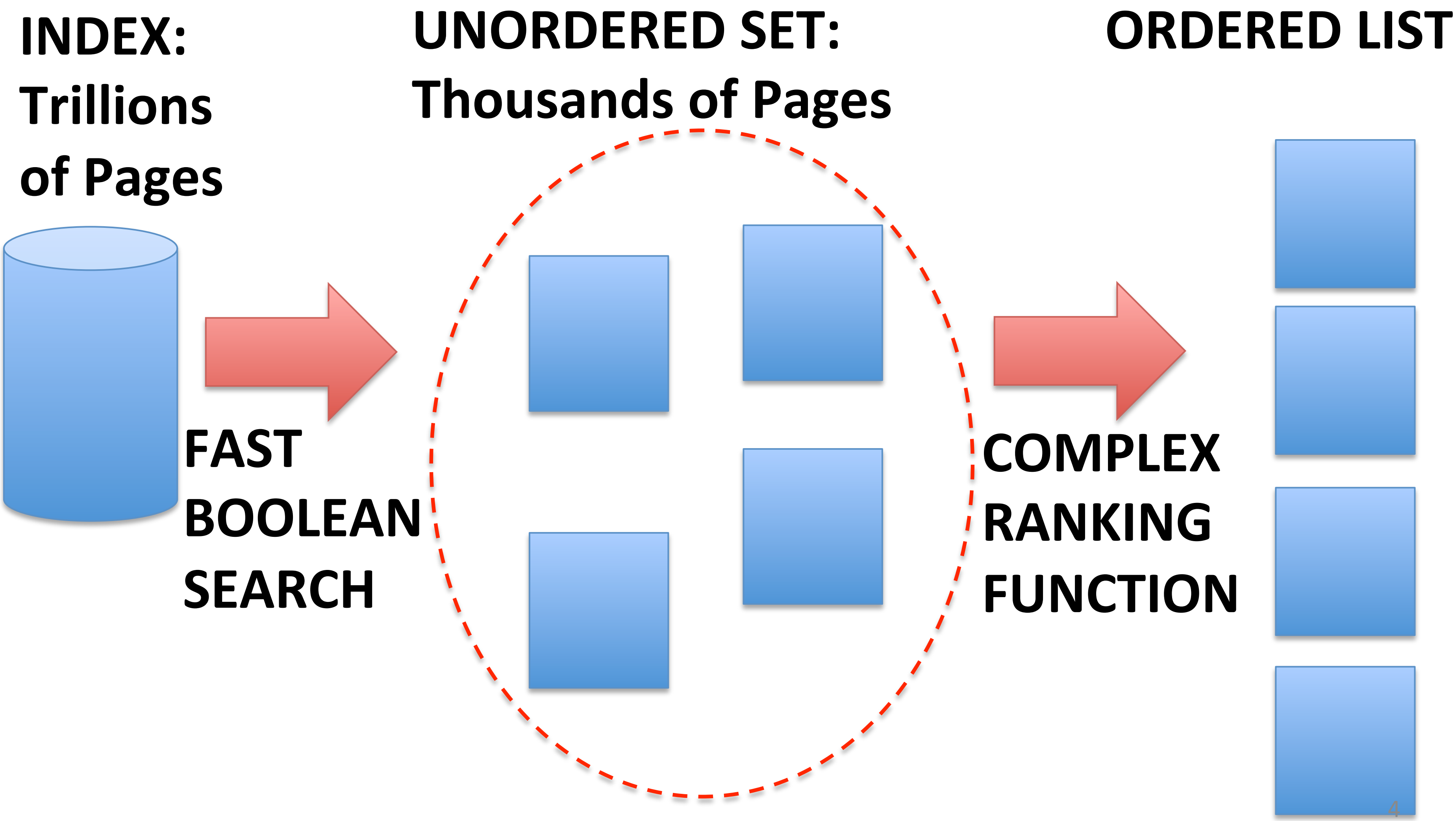
# What is Learning to Rank?



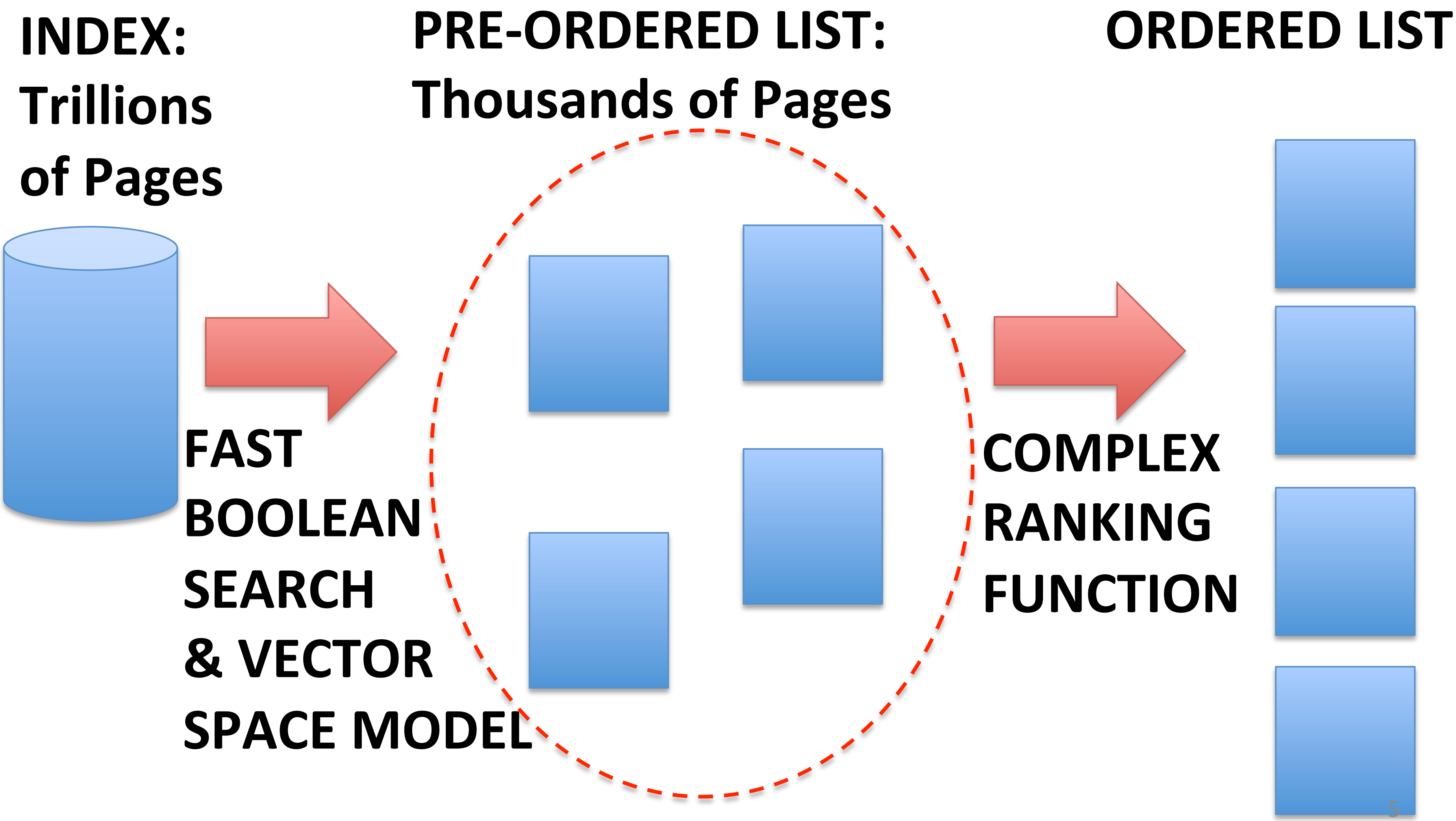
# Common in Search Engines



# Anatomy of a Search Engine



# Anatomy of a Search Engine



# Motivation

- In search engines, only the top results matter
- Machine learning approach:
  - Enables more **features** (signal sources)
  - Improves over Boolean search, vector space models like tf-idf

# Features of a top search result?

jhu cs courses 2016

**URL match?**

**Word match?**

**Popular page?**

**Recently updated?**

**User intention match?**

**Not spam?**

**Clickthrough log?**

**Johns Hopkins University Computer Science | Facebook**
https://www.facebook.com/compscijhu
**Johns Hopkins University Computer Science**, Baltimore. 407 likes · 5 talking about this. **Computer Science** at **Johns Hopkins University** (**CS@JHU**) is a...

600.318/418: Operating Systems - **Johns Hopkins University**
srl.**cs**.**jhu**.edu/**courses**/600.418/index.html
**Computer science** majors and graduate students will be admitted regardless of enrollment limits. ... This **course** provides an introduction to operating systems.

Department of **Computer Science | Course Information**
www.**cs**.**jhu**.edu/**course**-info
**CS Course** Catalog – complete list of departmental **courses** with descriptions. Note that not all **courses** are offered every year. **Course** Area Designators – a chart ...

**Johns Hopkins University • Free Online Courses** and ...
www.class-central.com › Universities › **Johns Hopkins University**
Discover free online **courses** taught by **Johns Hopkins University**. Watch videos, do assignments, earn a certificate while learning from some of the best Professors.

# Useful Features

- Based on Query (q) and Document (d)
  - Various Boolean search and vector space model results, applied to document text, URL, title
  - Click-through: e.g. How many times d is clicked given q vs. How many times d is skipped
  - Results after Query-expansion
- Based on Document only (static)
  - Popularity of page: #Likes, #inlinks, Pagerank
  - Domain structure: main page or subpage
- Many, many more

# Re-cap: Goal is to improve top ranked results via many features

jhu cs courses 2016

URL match?

Recently updated?

Word match?

Popular page?

User intention match?

Not spam?

Clickthrough log?

**Johns Hopkins University Computer Science | Facebook**
https://www.facebook.com/compscijhu
**Johns Hopkins University Computer Science**, Baltimore. 407 likes · 5 talking about this. **Computer Science** at **Johns Hopkins University** (**CS@JHU**) is a...

600.318/418: Operating Systems - **Johns Hopkins University**
srl.**cs.jhu**.edu/**courses**/600.418/index.html
**Computer science** majors and graduate students will be admitted regardless of enrollment limits. ... This **course** provides an introduction to operating systems.

Department of **Computer Science** | **Course Information**
www.**cs**.**jhu**.edu/**course**-info
**CS Course** Catalog – complete list of departmental **courses** with descriptions. Note that not all **courses** are offered every year. **Course** Area Designators – a chart ...

**Johns Hopkins University** • **Free Online Courses** and ...
www.class-central.com › Universities › **Johns Hopkins University**
Discover free online **courses** taught by **Johns Hopkins University**. Watch videos, do assignments, earn a certificate while learning from some of the best Professors.

# Challenge: How to weight features?

- If there's only a few, we can manually tune.

  e.g. score = 3 x #wordmatch + 1 x #inlink

- If there are many, we rely on machine learning (i.e. Learning to Rank)

# Summary so far

1. Ranking: Unordered set → Ordered list
2. Search engines: only top results matter
3. Good ranking requires many features
4. Next: how to weight features

# Problem Formulation


<u>1. Feature extraction</u>

<query, $doc_1$> → vector $x_1$

<query, $doc_2$> → vector $x_2$

<query, $doc_3$> → vector $x_3$

<u>2. Apply ranking function & sort</u>

$F(x_1) = 3$ → Rank 2

$F(x_2) = 1$ → Rank 3 (worst)

$F(x_3) = 4$ → Rank 1 (best)

<u>What function class for F()?</u>

Assume linear weights: $F(x_i) = w^T x_i$

Learn weights w that replicate ranking on training set

# Training Set

| Query 1 | Labels for each query-doc pair |
| --- | --- |
| <query, $doc_1$> → vector $x_1$ | label($x_1$) = 3 |
| <query, $doc_2$> → vector $x_2$ | label($x_2$) = 1 |
| <query, $doc_3$> → vector $x_3$ | label($x_3$) = 4 |

| Query 2 | Labels for each query-doc pair |
| --- | --- |
| <query, $doc_1$> → vector $x_1$ | label($x_1$) = 3 (Very relevant) |
| <query, $doc_2$> → vector $x_2$ | label($x_2$) = 2 (Relevant) |
| <query, $doc_3$> → vector $x_3$ | label($x_3$) = 1 (Slightly relevant) |
| <query, $doc_4$> → vector $x_4$ | label($x_4$) = 0 (Irrelevant) |

# Where does the label come from?

- Human annotation
  - High quality, but expensive

- Click-through logs
  - Noisy, but cheap/abundant

# Notation

query: $q^{(n)} \qquad n = 1, \ldots, N$

document for query n: $d_i^{(n)} \qquad i = 1, \ldots, I_n$

vector of $D$ features per query-doc: $x_i^{(n)} \in \mathcal{R}^D$

label for each query-doc pair: $l_i^{(n)} \in \mathcal{Z}$

Training set: $\{q^{(n)}, \{x_i^{(n)}, l_i^{(n)}\}\}$

Ranking Function: $F(x_i^{(n)}) = w^T x_i^{(n)}$

# Different training approaches

- How to optimize something on a set with a sort operation? Reduce to traditional regression/classification problems

| Training Approach | Reduction |
| --- | --- |
| Point-wise | Document |
| Pair-wise | Two Documents |
| List-wise | All Documents per query |

# Point-wise Approach

Training set: $\{q^{(n)}, \{x_i^{(n)}, l_i^{(n)}\}\}$

Ranking Function: $F(x_i^{(n)}) = w^T x_i^{(n)}$

Find w that makes each F(x) equal to its label

Training Objective: $\sum_n \sum_i (F(x_i^{(n)}) - l_i^{(n)})^2$

Training Objective: $\sum_n \sum_i (F(x_i^{(n)}) - l_i^{(n)})^2$

$\rightarrow \sum_z (F(x_z) - l_z)^2$ where z ranges over all i, n

Solve with linear regression!

# Pair-wise Approach

Training set: $\{q^{(n)}, \{x_i^{(n)}, l_i^{(n)}\}\}$

Ranking Function: $F(x_i^{(n)}) = w^T x_i^{(n)}$

Find w that gives every pair the correct ranking

Training Objective:

$$F(x_i^{(n)}) > F(x_j^{(n)}) \quad \forall \quad i, j \quad \text{s.t.} \quad l_i^{(n)} > l_j^{(n)}$$

Training Objective:

$$F(x_i^{(n)}) > F(x_j^{(n)}) \quad \forall \quad i, j \quad \text{s.t.} \quad l_i^{(n)} > l_j^{(n)}$$
$$\rightarrow F(x_i^{(n)}) - F(x_j^{(n)}) > 0 \quad \forall \quad i, j \quad \text{s.t.} \quad l_i^{(n)} > l_j^{(n)}$$
$$\rightarrow w^T x_i^{(n)} - w^T x_j^{(n)} > 0 \quad \forall \quad i, j \quad \text{s.t.} \quad l_i^{(n)} > l_j^{(n)}$$
$$\rightarrow w^T (x_i^{(n)} - x_j^{(n)}) > 0 \quad \forall \quad i, j \quad \text{s.t.} \quad l_i^{(n)} > l_j^{(n)}$$
$$\rightarrow w^T (\delta_{ij}^{(n)}) > 0 \quad \forall \quad i, j \quad \text{s.t.} \quad l_i^{(n)} > l_j^{(n)}$$

Solve with binary classification!

Make a new sample out of every pair

Give new label: Positive for i,j pairs

Negative for j,i pairs

# Disclaimer

- We’ve focused on very simple ranking functions (linear) for simplicity
- In practice, more complex functions (e.g. decision trees, neural nets) are common
- Recommend further reading:
  - Dawei Yin, et. al. “Ranking Relevance in Yahoo Search”, Proceedings of KDD2016

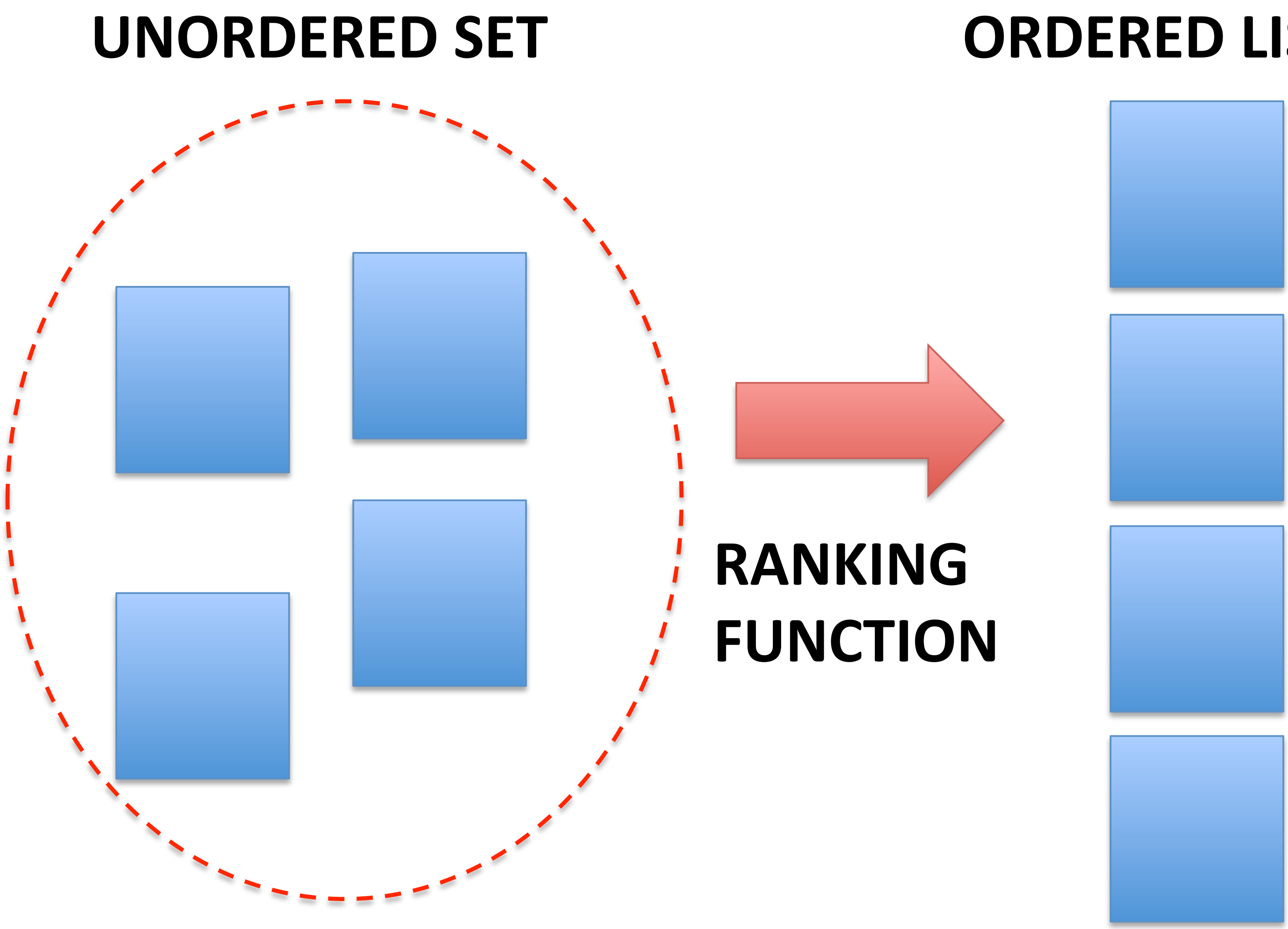
# Ranking Relevance in Yahoo Search

Dawei Yin[†], Yuening Hu[†], Jiliang Tang[†], Tim Daly Jr., Mianwei Zhou, Hua Ouyang, Jianhui Chen,
Changsung Kang, Hongbo Deng, Chikashi Nobata, Jean-Marc Langlois, Yi Chang[†]
Relevance Science, Yahoo! Inc.
[†]{daweiy,ynhu,jlt,yichang}@yahoo-inc.com

| query | methods | DCG1 | DCG3 | DCG5 |
|---|---|---|---|---|
| all | LogisticRank | 4.31 | 7.74 | 9.78 |
| | GBRank | 4.26 (-1.19%) | 7.52 (-2.81%)* | 9.51 (-2.81%)* |
| | LambdaMart | 4.24 (-1.60%) | 7.36 (-4.84%)* | 9.21 (-5.83%)* |
| top | LogisticRank | 5.69 | 9.67 | 12.08 |
| | GBRank | 5.56 (-2.22%) | 9.25 (-4.29%)* | 11.51 (-4.67%)* |
| | LambdaMart | 5.59 (-1.72%) | 9.08 (-6.04%)* | 11.02 (-8.75%)* |
| torso | LogisticRank | 3.88 | 7.23 | 9.26 |
| | GBRank | 3.88 (-1.77%) | 7.065 (-2.30%)* | 9.08 (-2.03%)* |
| | LambdaMart | 3.81 (-1.88%) | 6.97 (-3.64%)† | 8.92 (-3.64%)* |
| tail | LogisticRank | 2.91 | 5.65 | 7.16 |
| | GBRank | 2.99 (3.06%) | 5.65 (0.01%) | 7.19 (0.37%) |
| | LambdaMart | 2.88 (-0.71%) | 5.42 (-4.15%)† | 6.91 (-2.78%)† |

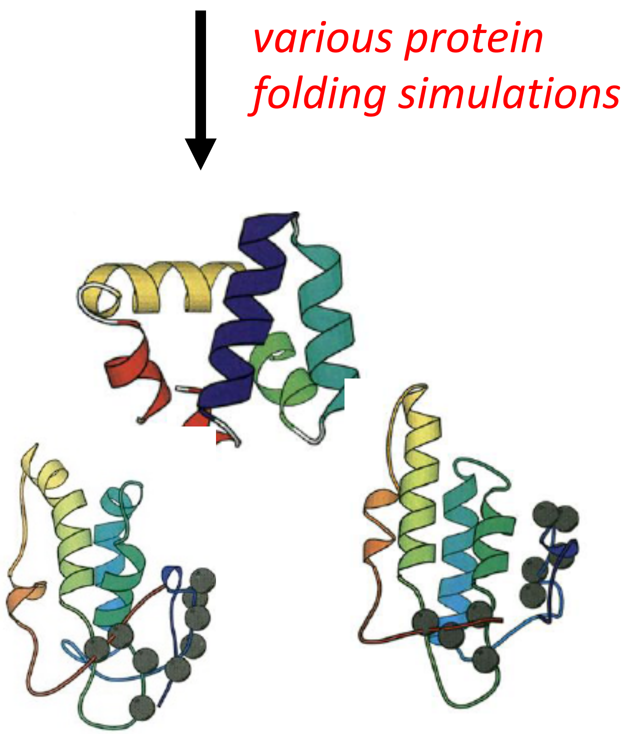
Table 1: Performance comparison of models using different learning algorithms. * denotes p-value<=0.01; † denotes p-value<=0.05.

# Other applications of Learning to Rank

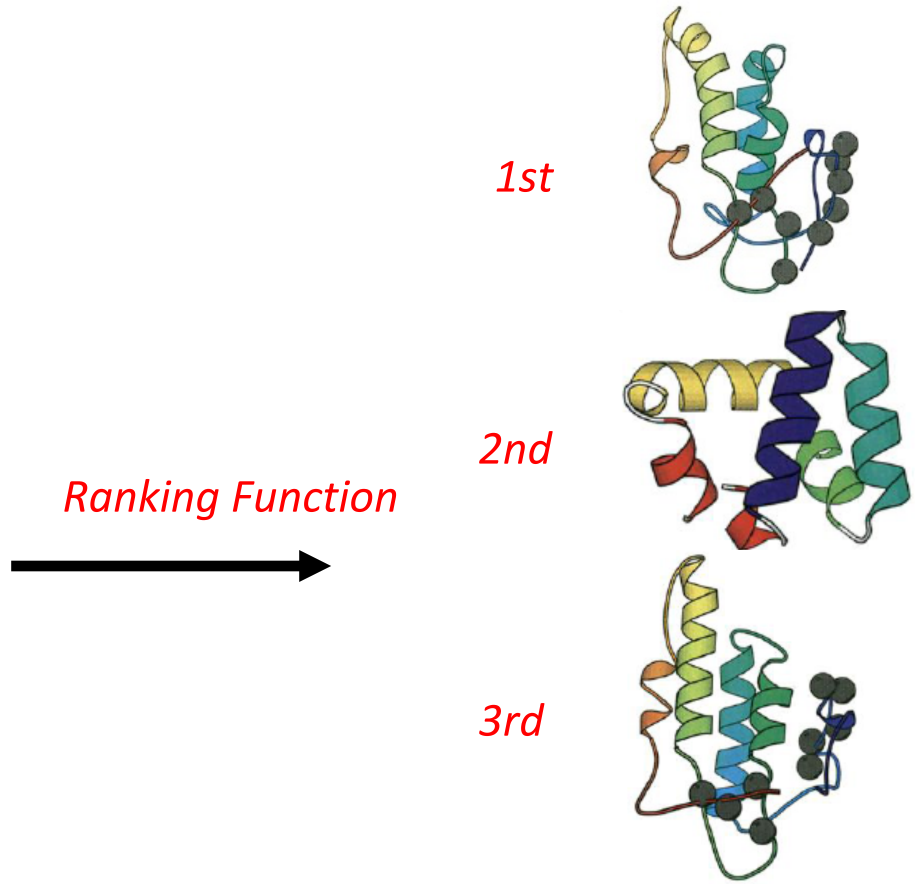


# Other applications: Protein structure prediction

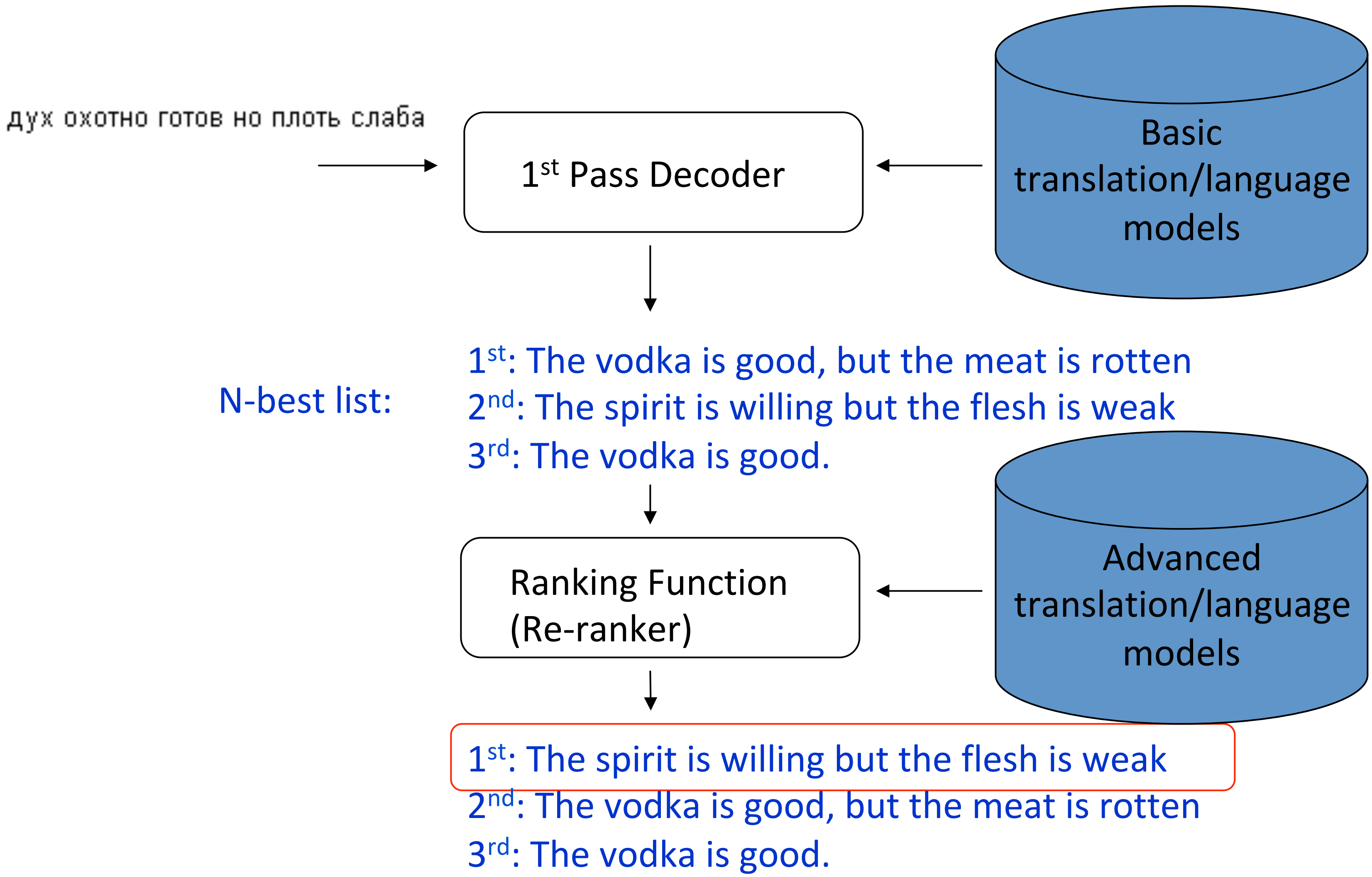
Amino Acid Sequence:
MMKLKSNQTRTYDGDGYKKRAACLCFSE




Candidate 3-D Structures



# Other applications: Machine Translation



# Summary

1. Ranking: Unordered set → Ordered list
2. Search engines: only top results matter
3. Good ranking requires many features
4. Approaches to learn weights of features (reduction to classification/regression)

# Lab: Build a learning-to-rank system

- https://github.com/kevinduh/tutorial-learning-to-rank