Ranking Evaluation, Ranking Systems (part 1)

HS Rankingaufgaben in der Computerlinguistik

Nils Reiter
nils.reiter@uni-koeln.de

Department of Digital Humanities

May 11, 2023
(Sommersemester 2023)

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Section 1

Ranking Evaluation

Terminology (Li, 2014)

$$Q : \{q_1, q_2, \ldots, q_n\}$$

q_j

Rank	Offering
1	o_1
2	o_6
3	o_5
4	o_1
5	o_2
6	o_4

q_i

o_5 o_6 o_4 o_2 o_1 o_3

Offerings $O = \{o_1, o_2, \ldots, o_n\}$

Rank	Offering
1	o_3
2	o_2
3	o_6
4	o_1
5	o_5
6	o_4

One Possible Ranking

Introduction

- Evaluation metrics for ranking tasks
- Comparison against a reference data set
- Ranked reference: $R \subset Q \times O^n$
  - I.e., for some queries, we know a "correct" ranking of length n
- Binary reference: $R \subset Q \times O$
  - I.e., for some queries, we know one or more "correct" objects
  - Metrics: Mean Reciprocal Rank (MRR), Precision at Position, Average Precision

Ranked Reference

Example

- Objects $O = \{o_1, o_2, o_3, \dots\}$
- Queries $Q = \{q_1, q_2, q_3, \dots\}$

Ranked Reference

Example

- Objects $O = \{o_1, o_2, o_3, \dots\}$
- Queries $Q = \{q_1, q_2, q_3, \dots\}$
- Reference data set $R = \{(q_1, \langle o_3, o_1, o_2 \rangle), (q_2, \langle o_2, o_1, o_3 \rangle), (q_3, \langle o_7, o_{12}, o_8 \rangle), \dots\}$

Ranked Reference

Example

- Objects $O = \{o_1, o_2, o_3, \dots\}$
- Queries $Q = \{q_1, q_2, q_3, \dots\}$
- Reference data set $R = \{(q_1, \langle o_3, o_1, o_2 \rangle), (q_2, \langle o_2, o_1, o_3 \rangle), (q_3, \langle o_7, o_{12}, o_8 \rangle), \dots\}$
- System output $S = \{(q_1, \langle o_3, o_1, o_2 \rangle), \quad (q_2, \langle o_1, o_2, o_3 \rangle), \quad (q_3, \langle o_3, o_2, o_1 \rangle)\}$

Ranked Reference

Example

- Objects $O = \{o_1, o_2, o_3, \dots\}$
- Queries $Q = \{q_1, q_2, q_3, \dots\}$
- Reference data set $R = \{(q_1, \langle o_3, o_1, o_2 \rangle), (q_2, \langle o_2, o_1, o_3 \rangle), (q_3, \langle o_7, o_{12}, o_8 \rangle), \dots\}$
- System output $S = \{(q_1, \langle o_3, o_1, o_2 \rangle), \quad (q_2, \langle o_1, o_2, o_3 \rangle), \quad (q_3, \langle o_3, o_2, o_1 \rangle)\}$
- Intuition:
  - S_1 is better than S_2, S_2 is better than S_3

Ranked Reference

Example

- Objects $O = \{o_1, o_2, o_3, \ldots\}$
- Queries $Q = \{q_1, q_2, q_3, \ldots\}$
- Reference data set $R = \{(q_1, \langle o_3, o_1, o_2\rangle), (q_2, \langle o_2, o_1, o_3\rangle), (q_3, \langle o_7, o_{12}, o_8\rangle), \ldots\}$
- System output $S = \{(q_1, \langle o_3, o_1, o_2\rangle), \quad (q_2, \langle o_1, o_2, o_3\rangle), \quad (q_3, \langle o_3, o_2, o_1\rangle)\}$
- Intuition:
  - S_1 is better than S_2, S_2 is better than S_3
- Core problem: Quantify difference between two sorted lists
  - Then: Average over items

Kendall's Tau

- Concept: Concordant pair of objects o_i, o_j
- A pair is concordant in R and S, if the objects are sorted equally in both rankings
  - E.g., if o_i comes before o_j *in both*

Kendall's Tau

- Concept: Concordant pair of objects o_i, o_j
- A pair is concordant in R and S, if the objects are sorted equally in both rankings
  - E.g., if o_i comes before o_j *in both*

$$\tau = \frac{2c}{\binom{n}{2}} \quad c\text{: Number of concordant pairs}$$

Kendall's Tau

Example

$$
\begin{aligned}
R &= \{(q_1, \langle o_3, o_1, o_2\rangle), (q_2, \langle o_2, o_1, o_3\rangle), (q_3, \langle o_7, o_{12}, o_8\rangle), \dots\} \\
S &= \{(q_1, \langle o_3, o_1, o_2\rangle), \quad (q_2, \langle o_1, o_2, o_3\rangle), \quad (q_3, \langle o_3, o_2, o_1\rangle)\}
\end{aligned}
$$

τ_1

$$
\begin{aligned}
C &= \{(o_3, o_1), (o_3, o_2), \\
&\quad (o_1, o_2)\} \\
c &= 3 \\
\tau_1 &= \frac{3}{3}
\end{aligned}
$$

τ_2

$$
\begin{aligned}
C &= \{(o_2, o_3), (o_1, o_3)\} \\
c &= 2 \\
\tau_2 &= \frac{2}{3}
\end{aligned}
$$

τ_3

$$
\begin{aligned}
C &= \{\} \\
c &= 2 \\
\tau_3 &= \frac{0}{3}
\end{aligned}
$$

Section 2

Ranking Systems

Introduction

- Information retrieval (IR)
  - "Semantic Search"
  - Find information in a large collection of information bearers
  - E.g., find the document that contains the information we seek
- Prototypical application: Search engines

Introduction

- Information retrieval (IR)
  - "Semantic Search"
  - Find information in a large collection of information bearers
  - E.g., find the document that contains the information we seek
- Prototypical application: Search engines

Different eras

- Algorithmic / rule-based
- Learn to rank
  - Feature-based machine learning
  - Neural machine learning / deep learning

(Supervised) Machine Learning

Different Task Types

- Classification: Put objects into classes
  - E.g., “This text is a fantasy novel”

(Supervised) Machine Learning

Different Task Types

- Classification: Put objects into classes
  - E.g., "This text is a fantasy novel"
- Sequence labeling: Classification, but the objects are not independent of each other
  - E.g., "This word is a noun"

(Supervised) Machine Learning

Different Task Types

- Classification: Put objects into classes
  - E.g., “This text is a fantasy novel”
- Sequence labeling: Classification, but the objects are not independent of each other
  - E.g., “This word is a noun”
- Ordinal classification: Put objects into classes, but the classes have an order
  - E.g., “This review expresses a ★★★-opinion”

(Supervised) Machine Learning

Different Task Types

- Classification: Put objects into classes
  - E.g., “This text is a fantasy novel”
- Sequence labeling: Classification, but the objects are not independent of each other
  - E.g., “This word is a noun”
- Ordinal classification: Put objects into classes, but the classes have an order
  - E.g., “This review expresses a ★★★-opinion”
- Regression: Assign numbers to objects
  - E.g., “On this day, the temperature will be $25.5\,^{\circ}\mathrm{C}$”

Machine Learning

- Directly specifying conditions: Rule-based systems (no machine learning)
  - E.g., if the text has a wizard, it's a fantasy novel
- Machine learning
  - We provide a training examples
    - I.e., a data set for which we know 'correct' outcomes
  - During training, the model tries to learn conditions by itself
  - After training, the model can be applied to new (unseen) data objects
    - In research, we do mainly testing

Features

- Each object may be different
- How does the model generalizes from one object to the next?
- Objects are "translated" into features

Features

- Each object may be different
- How does the model generalizes from one object to the next?
- Objects are "translated" into features

Examples

- Binary feature: Does the word "wizard" appear in the text?
- Numeric feature: How often does the word "wizard" appear in the text?
- Categorical feature: Is the preceding word a verb, adjective or determiner?

Features

- Systems use hundreds or thousands of features
- Numeric features integrate well with most ML algorithms
- Features do not need to make sense for us humans
- Frequently used feature sets (for document-based learning)
  - "bag of words": Which words appear in which document?
  - "count vectors": Which words appear how often in each document?

Section 3

Rule-Based Ranking

Introduction

- Baseline system: Simple, used for comparison purposes
  - More advanced systems are structurally similar

demo

TF-IDF

Jones (1972)

- Very common idea on term weighting
- TF: Term frequency
  - How frequent is a term in a document?
- DF: Inverse document frequency
  - In how many documents does the term appear?

$$\mathrm{tfidf}(t, d) = \frac{\mathrm{tf}(t, d)}{\mathrm{df}(t)}$$

Section 4

Learn To Rank

Introduction

- Machine learning for ranking systems
- Supervised learning: Based on training data
  - Manually collected
  - Click-through data

Introduction

- Machine learning for ranking systems
- Supervised learning: Based on training data
  - Manually collected
  - Click-through data
- Important question: How to represent our learning task?

Ranked Reference

- Objects $O = \{o_1, o_2, o_3, \ldots\}$
- Queries $Q = \{q_1, q_2, q_3, \ldots\}$
- Reference data set $R = \{(q_1, \langle o_3, o_1, o_2 \rangle), (q_2, \langle o_2, o_1, o_3 \rangle), (q_3, \langle o_7, o_{12}, o_8 \rangle), \ldots\}$
- System output $S = \{(q_1, \langle o_3, o_1, o_2 \rangle), \quad (q_2, \langle o_1, o_2, o_3 \rangle), \quad (q_3, \langle o_3, o_2, o_1 \rangle)\}$

Learning Task

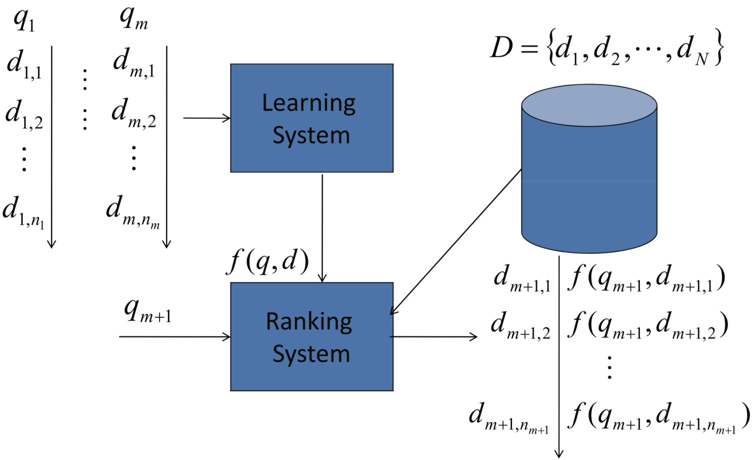

Figure: Learning to Rank for Document Retrieval (Li, 2014, 12)

Features

- Feature set needs to support generalization
- Learn to rank: "Features are defined as functions of query and [offering]" (Li, 2014, 13)

$$x_i = \phi(q_i, o_{i,j})$$

Training Procedure

$$
\begin{array}{l}
q_1 \left\{ \begin{array}{l} o_{1,1} \\ o_{1,2} \\ \vdots \\ o_{1,n_1} \end{array} \right. \\
\qquad \vdots \\
q_m \left\{ \begin{array}{l} o_{m,1} \\ o_{m,2} \\ \vdots \\ o_{m,n_m} \end{array} \right.
\end{array}
$$

Training Procedure

$$
\begin{array}{l}
q_1 \left\{ \begin{array}{l} o_{1,1} \\ o_{1,2} \\ \vdots \\ o_{1,n_1} \end{array} \right. \\
\quad\vdots \\
q_m \left\{ \begin{array}{l} o_{m,1} \\ o_{m,2} \\ \vdots \\ o_{m,n_m} \end{array} \right.
\end{array}
\quad \xrightarrow{\text{Labeling}} \quad
\begin{array}{l}
q_1 \left\{ \begin{array}{ll} o_{1,1} & y_{1,1} \\ o_{1,2} & y_{1,2} \\ \vdots & \\ o_{1,n_1} & y_{1,n_1} \end{array} \right. \\
\qquad\vdots \\
q_m \left\{ \begin{array}{ll} o_{m,1} & y_{m,1} \\ o_{m,2} & y_{m,2} \\ \vdots & \\ o_{m,n_m} & y_{m,n_m} \end{array} \right.
\end{array}
$$

Training Procedure

$$
\begin{array}{c}
q_1 \left\{ \begin{array}{l} o_{1,1} \\ o_{1,2} \\ \vdots \\ o_{1,n_1} \end{array} \right. \\
\vdots \\
q_m \left\{ \begin{array}{l} o_{m,1} \\ o_{m,2} \\ \vdots \\ o_{m,n_m} \end{array} \right.
\end{array}
\xrightarrow{\text{Labeling}}
\begin{array}{c}
q_1 \left\{ \begin{array}{ll} o_{1,1} & y_{1,1} \\ o_{1,2} & y_{1,2} \\ \vdots & \\ o_{1,n_1} & y_{1,n_1} \end{array} \right. \\
\vdots \\
q_m \left\{ \begin{array}{ll} o_{m,1} & y_{m,1} \\ o_{m,2} & y_{m,2} \\ \vdots & \\ o_{m,n_m} & y_{m,n_m} \end{array} \right.
\end{array}
\xrightarrow{\substack{\text{Feature} \\ \text{Extraction}}}
\begin{array}{c}
\left\{ \begin{array}{ll} x_{1,1} & y_{1,1} \\ x_{1,2} & y_{1,2} \\ \vdots & \\ x_{1,n_1} & y_{1,n_1} \end{array} \right. \\
\vdots \\
\left\{ \begin{array}{ll} x_{m,1} & y_{m,1} \\ x_{m,2} & y_{m,2} \\ \vdots & \\ x_{m,n_m} & y_{m,n_m} \end{array} \right.
\end{array}
$$

Training Procedure

Beispiel Pairwise

x = [0.5, 0, 0.73, 0.1, 1, -0.3]

y = "Vektor 1 ist vorne" / = 0

Beispiel Pointwise

x = [0.5, 0, 0.73] y = 0.1

$$
\begin{array}{c}
q_1 \left\{ \begin{array}{l} o_{1,1} \\ o_{1,2} \\ \vdots \\ o_{1,n_1} \end{array} \right. \\
\vdots \\
q_m \left\{ \begin{array}{l} o_{m,1} \\ o_{m,2} \\ \vdots \\ o_{m,n_m} \end{array} \right.
\end{array}
\xrightarrow{\text{Labeling}}
\begin{array}{c}
q_1 \left\{ \begin{array}{ll} o_{1,1} & y_{1,1} \\ o_{1,2} & y_{1,2} \\ \vdots & \\ o_{1,n_1} & y_{1,n_1} \end{array} \right. \\
\vdots \\
q_m \left\{ \begin{array}{ll} o_{m,1} & y_{m,1} \\ o_{m,2} & y_{m,2} \\ \vdots & \\ o_{m,n_m} & y_{m,n_m} \end{array} \right.
\end{array}
\xrightarrow{\text{Feature Extraction}}
\begin{array}{c}
\left\{ \begin{array}{ll} x_{1,1} & y_{1,1} \\ x_{1,2} & y_{1,2} \\ \vdots & \\ x_{1,n_1} & y_{1,n_1} \end{array} \right. \\
\vdots \\
\left\{ \begin{array}{ll} x_{m,1} & y_{m,1} \\ x_{m,2} & y_{m,2} \\ \vdots & \\ x_{m,n_m} & y_{m,n_m} \end{array} \right.
\end{array}
\xrightarrow{\text{Learning}} f(x)
$$

Learning Approaches

Pointwise

- Learning model predicts rank/score for individual pair (x_i, y_i)
- Typical supervised learning $f(x) = y$
  - If y class label: classification
  - If y real number: regression
  - If y graded label: ordinal classification

Learning Approaches

Pointwise

- Learning model predicts rank/score for individual pair (x_i, y_i)
- Typical supervised learning $f(x) = y$
  - If y class label: classification
  - If y real number: regression
  - If y graded label: ordinal classification
- Problem reduced to base task type
- Standard algorithms available

Learning Approaches

Pairwise

- Model predicts an order between two feature vectors
- $f(x_i, x_j) = y$
  - Classification: $y \in \{x_i \prec x_j, x_j \prec x_i\}$
    ($a \prec b$ expresses that a comes before b in the ranking)
  - Regression: $y \in [0; 1]$
    (higher number comes first in ranking)

Learning Approaches

Listwise

- Model predicts an order for a set of feature vectors
- Most natural way
- No standard ML problem
  - "a new problem for machine learning and conventional techniques in machine learning cannot be directly applied" (Li, 2014, 27)
- $$f\left(\left\{\begin{matrix} x_1, \\ x_2, \\ \vdots \\ x_n \end{matrix}\right\}\right) = \left\{\begin{matrix} s_{x_1}, \\ s_{x_2}, \\ \vdots \\ s_{x_n} \end{matrix}\right\}$$

Section 5

Summary

Summary

- Evaluation if ranked reference data: Kendall's Tau
  - Defined over concordant pairs of objects
- Ranking Systems
  - Rule-based / algorithmic
    - Frequency ranking
    - TF-IDF
  - Learn to rank
    - Instance: Pair of query and offerening
  - Pointwise: Predict a score for each pair
  - Pairwise: Predict which one of two instances comes first
  - Listwise: Genuin ranking

References I

Jones, Karen Spärck (1972). "A Statistical Interpretation of Term Specificity and its Application in Retrieval". In: *Journal of Documentation* 28.1, pp. 11–21. DOI: 10.1108/eb026526.

Li, Hang (2014). *Learning to Rank for Information Retrieval and Natural Language Processing*. Ed. by Graeme Hirst. 2nd ed. Synthesis Lectures on Human Language Technologies. Morgan & Claypool.