Utilizing Machine Learning in Information Retrieval:

• Text Classification
• Learning to Rank

(COSC 488)
Nazli Goharian
nazli@cs.georgetown.edu

Literatures used to prepare the slides: See last page!

What is Text Classification?

Text classification also known as text categorization, topic classification, or topic spotting is the process of assigning predefined categor(ies)/topic(s)/class(e)s/label(s) to a document that reflect its overall contents.
Application of Text Classification

- News Classification
  - “Politics”, “Sports”, “Business”

- Shopping Products Classification
  - “Electronics”, “Home Appliances”, “Books”
Tropical storms are building up in the south Pacific due to high pressure belts. The rains may continue for few more days.

Users interested in weather news (standing queries)

Application of Text Classification

• News Routing/Filtering

• Spam Filtering
  – “Spam”, “Not Spam”
Improving Search Results via Text Classification

• Query is searched in the user selected categories in web directories

• Categorized result set is presented to user

• Learning to rank -- (more recent efforts)
  Using various document features such as document length, age, etc. and their relevance to a query, build a model to rank/re-rank the documents

• Query category is searched against categorized pages
  (vertical search, advertisement search,…)

Web Directories

Constructing Web directories to be able to browse information via predefined set of categories:
• Yahoo
• dmoz Open Directory Project (ODP)
• Existing directories are based on human efforts
  • 80,000 editors involved to maintain ODP; www.dmoz.org

Using Web directories (Yahoo,ODP, Wikipedia,…) as training data, the classifier classifies new web pages into categories
Supervised Learning (Classification)

- Learning a model (classifier), using annotated training samples (documents) to classify any new incoming document into pre-defined set of topics
- Each Training document has one/more label(s)
- Various learning algorithms exists, examples:
  - Example: Naïve Bayes, decision tree, support vector machine, neural network, regression, K-nearest neighbor,...
- Model/Classifier is used to classify incoming (test) documents
Example: Single-labeled Document

The Dow Jones industrial average lost 26 points, or 0.3%. The S&P 500 index fell 6 points, or 0.6%. The Nasdaq composite was little changed. Stocks slipped through most of the session as investors mulled the implications of a weaker-than-expected reading on the services sector of the economy, and mixed reports on the jobs market, ahead of Friday's big monthly payrolls report.


Example: Multi-labeled Document

President Obama, in his proposed 2011 budget, is calling on Congress to make a number of tax changes for individuals. Some ideas are new. Many others were made last year, but not enacted by Congress. So the estimates of the revenue that may be raised by his proposals may be overly optimistic.

Source: CNN (http://money.cnn.com/2010/02/01/pf/taxes/obama_budget_tax_changes/index.html)
Hard Categorization vs. Ranking Categorization

Hard Categorization
Complete decision of True or False for each pair \( <d_j, c_i> \)

<table>
<thead>
<tr>
<th>Document</th>
<th>Category Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>d_1</td>
<td>c_1, c_2</td>
</tr>
<tr>
<td>d_2</td>
<td>c_2</td>
</tr>
<tr>
<td>d_3</td>
<td>c_2, c_4</td>
</tr>
<tr>
<td>d_4</td>
<td>c_3</td>
</tr>
</tbody>
</table>

Ranking (Soft) Categorization
Given \( d_j \in D \), rank the categories according to their estimated appropriateness to \( d_j \)

<table>
<thead>
<tr>
<th>Document</th>
<th>Category</th>
<th>Estimated appropriateness</th>
</tr>
</thead>
<tbody>
<tr>
<td>d_1</td>
<td>c_2</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>c_1</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>c_3</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>c_4</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Types of Classification
from: X. Qi and B. Davison, ACM Computing Surveys, 2009

Binary

Hard, Multi-class, Single-label

Hard, Multi-class, Multi-label

Soft, Multi-class

Classifier

Class C \( \text{OR} \) Class C

Class C_1 \( \text{OR} \) Class C_2 \( \text{OR} \) … Class C_n

Class C_1 \( \text{and} \) Class C_3

Class C_i: 0.2, C_2: 0.2, …………..C_n: 0.6,
Feature Selection

- **Feature Selection** in text classification refers to selecting a subset of the collection terms and utilize them in the process of text classification.

- Good features are better indicators of a class label

- Feature reduction tends to:
  - Reduce *overfitting*
  - Improve performance due to reducing dimensionality

- **Feature Extraction** provides more detailed features and feature relationships *(not covered in this course)*
Feature Selection

• Given a feature set $X = \{x_i \mid i=1\ldots N\}$, find a subset $Y = \{x_{i_1}, x_{i_2}, \ldots, x_{i_M}\}$, with $M<N$, that increases the probability of correct classification.

Text Features

• Feature space in text may include:
  – Lexical features (words, phrases)
  – Part-of-Speech (POS)
  – N-grams
  – Synonyms
  – …. 

• General feature types may be:
  – Numeric
  – Nominal
  – Ordinal
  – Ratio
Web Page Features

- **Additional features** are utilized in Web page classification task:
  - On-Page Features
  - Neighboring Page Features (External Links)

On-Page Features

**HTML tags:**
- title
- headings
- metadata
- main text

HTML tags usually removed in pre-processing; the content of tags preserved

URL – classify without using page content
Neighboring-Page Features

- Neighbors (linked pages) have similar topics and categories
- Number of steps from a page -- shown as 2 (parent, child, sibling, grand parent, grand child); more steps more expensive & less effective
- Although all useful, but sibling is shown to be more effective
- Using only portion of neighboring content: title, anchor text, text closer to hyperlink to train a classifier
- Voting -- majority class of neighbors used

(from X. Qi and B. Davison, ACM Computing Surveys, 2009)
Context Features of a Document

- Author
- Weblog Article
- Location
- Time
- Source
- Communities

Feature Selection Algorithms

- Frequency based FS:
  - df
  - tf-idf
  - Tf-icf

- Commonly used Information Theoretic based FS:
  - Mutual Information
  - Information Gain
  - $\chi^2$ Statistic (CHI)
  - Odds Ratio

(Note: There are some more FS algorithms!)
Feature Selection: Frequency based

- **DF** (Document Frequency): *Frequency of a term in the collection*
  - Retain terms that are not *stop terms* (high df) and do not have very low df (noise, not of interest)

- **TF-IDF**
  - *tf*: frequency of a term in a document -- commonly normalized
  - *idf*: inverse document frequency
    
    \[
    \text{tfidf}(t_i, d_j) = TF(t_i, d_j) \cdot \log \left( \frac{D}{df(t_i)} \right)
    \]
  - Retain terms with high *tf-idf* in a document

- **TF-ICF**
  - Analogous to *tf-idf* but considering the frequency of term in the category.
    
    \[
    \text{tficf}(t_i, c_j) = TF(t_i, c_j) \cdot \log \left( \frac{C}{cf(t_i)} \right)
    \]

---

Feature Selection (FS)

Consider the Term-Class incidence table:

<table>
<thead>
<tr>
<th>Case</th>
<th>Does in class: ( c_p )</th>
<th>Does not in class: ( \bar{c}_p )</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does that contain term ( k_i )</td>
<td>( n_{k,p} )</td>
<td>( n_i - n_{k,p} )</td>
<td>( n_i )</td>
</tr>
<tr>
<td>Does that do not contain term ( k_i )</td>
<td>( n_p - n_{k,p} )</td>
<td>( N_i - n_i - (n_p - n_{k,p}) )</td>
<td>( N_i - n_i )</td>
</tr>
<tr>
<td>All docs</td>
<td>( n_p )</td>
<td>( N_i - n_p )</td>
<td>( N_i )</td>
</tr>
</tbody>
</table>

The notations used in this table are used in the FS algorithms of the next few pages!

*From: Modern Information retrieval, R. Baeza-Yates & B. Ribeiro-Neto, 2011*
FS: Mutual Information (MI)

Measuring the amount of information the presence of a term contributes to the classification, MI between term $k_i$ and set of classes $C$ is expressed as expected value of:

$$I(k_i, c_p) = \log \frac{P(k_i, c_p)}{p(k_i) P(c_p)} = \log \left( \frac{\frac{n_{i,p}}{N_p}}{\frac{n_i}{N_t} \frac{n_{p}}{N'}} \right)$$

Two alternates: 1) across all classes; 2) maximum term information:

$$MI(k_i, C) = \sum_{p=1}^{c} p(c_p) I(k_i, c_p) = \sum_{p=1}^{c} \frac{n_{i,p}}{N_t} \log \left( \frac{\frac{n_{i,p}}{N_p}}{\frac{n_i}{N_t} \frac{n_{p}}{N'}} \right)$$

$$I_{\text{max}}(k_i, C) = \max_{p=1}^{c} I(k_i, c_p) = \max_{p=1}^{c} \log \left( \frac{\frac{n_{i,p}}{N_p}}{\frac{n_i}{N_t} \frac{n_{p}}{N'}} \right)$$

FS: Information Gain (IG)

Measuring the amount of information both the presence and the absence of a term contribute to the classification. Terms with $IG \geq \text{threshold}$ are kept.

$$IG(k_i, C) = -\sum_{p=1}^{l} P(c_p) \log P(c_p)$$

$$-(- \sum_{p=1}^{l} P(c_p, k_i) \log P(c_p | k_i))$$

$$-(- \sum_{p=1}^{l} P(c_p, \bar{k}_i) \log P(c_p | \bar{k}_i))$$

$$IC(k_i, C) = -\sum_{p=1}^{l} \left( \frac{n_{i,p}}{N_t} \log \frac{n_{i,p}}{N_t} \right) - \left( \frac{n_{i,p}}{N_t} \log \frac{n_{i,p}}{N_t} \right) - \left( \frac{n_{i,p} - n_{i,p}}{N_t - n_i} \log \frac{n_{i,p} - n_{i,p}}{N_t - n_i} \right)$$
FS: Chi Square ($\chi^2$)

- Chi Square measures the dependency between the term and the class (value of zero indicates independency)

$$\chi^2(k_i, c_p) = \frac{N_i \left( P(k_i, c_p) P(k_i, c_p) - P(k_i, c_p) P(k_i, c_p) \right)}{P(c_p) P(c_p) P(k_i) P(k_i)}$$

- Calculate $\chi^2$ of a term over all categories and retain the term if the value meets a threshold. Two alternatives:

1) Averaging over all categories: $\chi^2_{\text{avg}}(k_i) = \sum_{p=1}^{L} P(c_p) \chi^2(k_i, c_p)$

2) Considering the largest value: $\chi^2_{\text{max}}(k_i) = \max_{p=1}^{L} \chi^2(k_i, c_p)$

---

FS: Chi Square ($\chi^2$) (Cont’d)

- Chi Square measures the dependency between the term and the class (value of zero indicates independency)

$$\chi^2(k_i, c_p) = \frac{N_i \left( P(k_i, c_p) P(k_i, c_p) - P(k_i, c_p) P(k_i, c_p) \right)^2}{P(c_p) P(c_p) P(k_i) P(k_i)}$$

$$\chi^2(k_i, c_p) = \frac{N_i \left( (n_{i,p} - n_i - n_p + n_{i,p}) \right)^2}{n_p (N_i - n_p) n_i (N_i - n_i)}$$

$$= \frac{N_i \left( n_{i,p} N_i - n_p n_i \right)^2}{n_p n_i (N_i - n_p) (N_i - n_i)}$$
FS: Odds Ratio

- Odds Ratio reflects the odds of the word occurring in the **positive** class normalized by that of the **negative** class.

- Odds Ratio for a term $t_k$ in category $c_i$

\[
OR(t_k, c_i) = \frac{P(t_k | c_i)[1 - P(t_k | \overline{c_i})]}{P(t_k | \overline{c_i})[1 - P(t_k | c_i)]}
\]

Supervised Learning Algorithms

- Naïve Bayes
- K-Nearest Neighbor (KNN)
- Support Vector Machines (SVM)
- Decision-tree
- Decision-Rule classifiers
- Neural Networks
- Rocchio
- HMM
- CRF

*Only these two are covered in this course!*
Representation of Text

This week, the United Nations created the position of czar in the global fight against a possible avian influenza pandemic. Meanwhile, officials here in the United States acknowledged the country is unprepared if this never-before-seen strain of flu, known to scientists as H5N1 virus, were to hit this winter.

- Commonly used pre-processing: stop word removal, stemming,…

\[ d1 : \langle \text{week, united, nations, create, position, czar, global, fight, against, possible,.....} \rangle \]

<table>
<thead>
<tr>
<th>Term</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week</td>
<td>1</td>
</tr>
<tr>
<td>united</td>
<td>2</td>
</tr>
<tr>
<td>nation</td>
<td>1</td>
</tr>
<tr>
<td>..........</td>
<td>..........</td>
</tr>
</tbody>
</table>

Phrases:
- United nations
- Avian influenza
- ..........

Naïve Bayes Text Classifier

- Text as “bag-of-words”
- Independent assumption -- occurrence of terms and their positions

- Building Model:
  - For each category \( c_i \) build a probabilistic model

\[ P(T : t_1, t_2, \ldots t_n \mid c_i) \]

- Calculate the prior probability \( P(C_1) \)
Naïve Bayes Text Classifier

• Classify Text:
  – Calculate probability of each category for a given text
    \[ P(c_i \mid d_j) = p(c_i)P(d_j \mid c_i) \]

  – The category \( c_i \) with the highest score among all categories \( C \) is the one that is most probable to generate the text \( d_j \)
    \[ c_{\text{max a posteriori}} = \arg \max_{c_i \in C} p(c_i)P(d_j \mid c_i) \]

Naïve Bayes Text Classifier

\[
P(c_i \mid d_j) = p(c_i) \underbrace{P(d_j \mid c_i)}_{\sum_{i=1}^{r} \log P(t_{kj} \mid c_i)}
\]

\[
\prod_{k=1}^{l} P(t_{kj} \mid c_i) = \sum_{i=1}^{r} \log P(t_{kj} \mid c_i)
\]
Naïve Bayes Text Classifier

- Need to estimate the probability: \( P(t_{kj} \mid c_i) \)
  - Multinomial model:
    \[
    \frac{\text{number of times term } t_{kj} \text{ appears in category } c_i + 0.5}{\text{total terms in } c_i + 1}
    \]
  - Binomial or Bernoulli model:
    \[
    \frac{\text{number of documents in category } c_i \text{ that term } t_{kj} \text{ appears}}{\text{total documents in } c_i}
    \]

\[
\begin{align*}
\log \left( \frac{\text{docs in } c_i}{\text{total docs}} \right) & \prod_{k=1}^{T} P(t_{kj} \mid c_i) = \sum_{i=1}^{T} \log P(t_{kj} \mid c_i) \\
& \text{number of times term } t_{kj} \text{ appears in category } c_i + 0.5 \text{ total terms in } c_i + 1
\end{align*}
\]

To avoid a zero if a new term appears → Smoothing
- Various approaches: Dirchelet prior, Laplace...
### Example

<table>
<thead>
<tr>
<th>Doc-1</th>
<th>Doc-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category: Computers</td>
<td>Category: Computers</td>
</tr>
<tr>
<td>The sales of laptops in 2009 was high as many OS were released</td>
<td>Many OS provide varying level of securities for laptops as they tend to switch networks. This makes the laptops more secure from computer viruses</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Doc-3</th>
<th>Doc-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category: Epidemic</td>
<td>Category: Epidemic</td>
</tr>
<tr>
<td>A new virus called H1N1 causes Swine Flu.</td>
<td>Bird flu is caused by a virus called H5N1. The disease is of concern to humans, who have no immunity against it.</td>
</tr>
</tbody>
</table>

---

### Example

Assume that **red** terms are the selected features:

<table>
<thead>
<tr>
<th>Doc-1</th>
<th>Doc-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category: Computers</td>
<td>Category: Computers</td>
</tr>
<tr>
<td>The sales of <strong>laptops</strong> in 2009 was high as many <strong>OS</strong> were released</td>
<td>Many <strong>OS</strong> provide varying level of securities for <strong>laptops</strong> as they tend to switch networks. This makes the <strong>laptops</strong> more secure from <strong>computer viruses</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Doc-3</th>
<th>Doc-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category: Epidemic</td>
<td>Category: Epidemic</td>
</tr>
<tr>
<td>A new <strong>virus</strong> called H1N1 causes <strong>Swine Flu</strong>.</td>
<td><strong>Bird flu</strong> is caused by a <strong>virus</strong> called H5N1. The <strong>disease</strong> is of concern to humans, who have no <strong>immunity</strong> against it.</td>
</tr>
</tbody>
</table>
Example: Naïve Bayes Text Classifier

**Task:** Classify D5: “A deadly *virus* called *H1N1* was detected in various parts of the world”

- $P(\text{Computers}|D5) = P(\text{Computers}) \cdot P(\text{Virus}|\text{Computers}) \cdot P(\text{H1N1}|\text{Computers})$

- $P(\text{Epidemic}|D5) = P(\text{Epidemic}) \cdot P(\text{Virus}|\text{Epidemic}) \cdot P(\text{H1N1}|\text{Epidemic})$

$P(\text{Epidemic}|D5) > P(\text{Computers}|D5)$

Thus, class of D5 is Epidemics

---

**Vector Space Classification**

- Documents represented as a vector with generally *tfidf* of terms

- Generally classification decisions are based on a similarity/distance measure
  - Centroids [averages] play a role

- Sample algorithms:
  - Rocchio
  - K Nearest Neighbor (kNN)
  - SVM

> *Only this one is covered in this course!*
Nearest Neighbor Classifiers

Slide from: Tan, Steinback, Kumar, 2004

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck

K-Nearest Neighbor Classifier

- No model is built (lazy learner) a priori
  (Classification done based on raw training data)
- The class of a document will be the class of the majority class of the $k$ nearest neighbor (majority voting)
- The relatedness/nearness of two documents can be quantified in terms of similarity (eg. Cosine measure) or distance (eg. Euclidean distance)
  - Different weight for different features
  - Feature values can be normalized to prevent different handling (may prefer different handling!)
- Sensitivity to value of $K$
  - Picked empirically, domain knowledge
Distance/Similarity Measures

Euclidean Distance:
\[
\text{dist}(d_i,d_j) = \sqrt{(|d_{i1} - d_{j1}|^2 + |d_{i2} - d_{j2}|^2 + \ldots + |d_{ip} - d_{jp}|^2)}
\]

Cosine Similarity:
\[
\text{Sim}(d_i,d_j) = \frac{\sum_{k=1}^{K} d_{ik} \times d_{jk}}{\sqrt{\sum_{k=1}^{K} (d_{ik})^2 \sum_{k=1}^{K} (d_{jk})^2}}
\]

Term weight:
\[
w_{ij} = \frac{(\log tf_{ij} + 1.0) \times idf_j}{\sum_{j=1}^{K} [(\log tf_{ij} + 1.0) \times idf_j]}\]

Evaluation Metrics

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class=Yes</td>
</tr>
<tr>
<td>Class=Yes</td>
<td>TP</td>
</tr>
<tr>
<td>Class=No</td>
<td>FP</td>
</tr>
</tbody>
</table>

Precision \((p)\) = \(\frac{tp}{tp+fp}\)

Recall \((r)\) = \(\frac{tp}{tp+fn}\)

F1- measure \((F1)\) = \(\frac{2rp}{r + p}\)
Macro-Averaging

- **Macro-average:**
  - Equal weight to each category

  \[
  \text{Macro - Precision} = \frac{\text{Precision}(A) + \text{Precision}(B) + \text{Precision}(C)}{3}
  \]

  \[
  \text{Macro - Recall} = \frac{\text{Recall}(A) + \text{Recall}(B) + \text{Recall}(C)}{3}
  \]

  \[
  \text{Macro - F1 Measure} = \frac{\text{F1 Measure}(A) + \text{F1 Measure}(B) + \text{F1 Measure}(C)}{3}
  \]

Micro-Averaging

- **Micro-average:**
  - Equal weight to each sample (record, document)

  \[
  \text{Micro - Precision} = \frac{TP_A + TP_B + TP_C}{TP_A + TP_B + TP_C + FP_A + FP_B + FP_C}
  \]

  \[
  \text{Micro - Recall} = \frac{TP_A + TP_B + TP_C}{TP_A + TP_B + TP_C + FN_A + FN_B + FN_C}
  \]

  \[
  \text{Micro - F1 Measure} = \frac{2 \times \text{Micro - Precision} \times \text{Micro - Recall}}{\text{Micro - Precision} + \text{Micro - Recall}}
  \]
Performance Factors

- Performance of a model may depend on other factors besides the learning algorithm:
  - Class distribution
  - Cost of misclassification
  - Size of training and test sets
  - Good coverage

Learning Curve

Learning curve shows how accuracy changes with varying sample size
10-fold cross validation

- Training data: 90%
- Test data: 10%
- Each run will result in a particular classification rate.
- Average the ten classification rates for a final 10-fold cross validation classification rate.

Evaluation Dataset

- Manual labeling needs excessive effort
- Available Web directory: Yahoo directory & dmoz ODP (Open Directory Project)
- Several other sources available – nowadays Wikipedia
- Problem– not one given benchmark!
- Not one given domain!
Some of the Text Classification Benchmark Datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>No. of documents</th>
<th>No. of Categories</th>
<th>Size of dataset</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuters 21578</td>
<td>21,578</td>
<td>108 Categories (we used top 10)</td>
<td>28 MB</td>
<td>News Articles</td>
</tr>
<tr>
<td>20 News Group</td>
<td>20,000</td>
<td>20 categories</td>
<td>61 MB</td>
<td>News Articles</td>
</tr>
<tr>
<td>WebKB</td>
<td>8,282</td>
<td>7 categories</td>
<td>43 MB</td>
<td>Web Pages (University websites)</td>
</tr>
<tr>
<td>OHSUMED</td>
<td>54,710 (Total)</td>
<td>4,308 (we used top 50)</td>
<td>382 MB</td>
<td>Bio-medical Documents</td>
</tr>
<tr>
<td>GENOMICS (TREC 05)</td>
<td>4.5 million (Total) 591,689 (Subset)</td>
<td>20,184 (we used top 50)</td>
<td>15.5 GB</td>
<td>Bio-medical Documents</td>
</tr>
</tbody>
</table>

More benchmark datasets exist!

Sample Dataset:
*20 Newsgroups Hierarchy*
Learning to Rank


Putting it all together (borrowed from:

Learning to Rank

- Retrieval models need tuning parameters
  - Not a trivial task
  - may lead to overfitting
- Not one retrieval model outcome may suffice for ranking, a combination maybe helpful
  - Thus, using ML to automatically
    - Tune parameters
    - Combine ranking features

“Learning-to-rank” methods are those ranking methods that use ML for ranking!

For a given query $q$, its related document $d$ is represented as a feature vector $x = \Phi(q, d)$

($\Phi$ is a feature extractor)

Typical features: qtf, BM25, PageRank, link info, …

- Learning process based on training data
  (training data is documents, user feedback, log, etc…..)
Learning to Rank: Sample Learning Features (Trec)

1. Term frequency (TF) of body
2. TF of anchor
3. TF of title
4. TF of URL
5. TF of whole document
6. Inverse document frequency (IDF) of body
7. IDF of anchor
8. IDF of title
9. IDF of URL
10. IDF of whole document
11. TF*IDF of body
12. TF*IDF of anchor
13. TF*IDF of title
14. TF*IDF of URL
15. TF*IDF of whole document
16. Document length (DL) of body
17. DL of anchor
18. DL of title
19. DL of URL
20. DL of whole document
21. BM25 of body
22. BM25 of anchor
23. BM25 of title
24. BM25 of URL
25. BM25 of whole document
26. LMIR.ABS of body
27. LMIR.ABS of anchor
28. LMIR.ABS of title
29. LMIR.ABS of URL
30. LMIR.ABS of whole document
31. LMIR.JM of body
32. LMIR.JM of anchor
33. LMIR.JM of title
34. LMIR.JM of URL
35. LMIR.JM of whole document
36. Sitemap based term propagation
37. Sitemap based score propagation
38. Hyperlink base score propagation: weighted in-link
39. Hyperlink base score propagation: weighted out-link
40. Hyperlink base feature propagation: uniform in-link
41. Hyperlink base feature propagation: uniform out-link
42. HITS authority
43. HITS hub
44. PageRank
45. HostRank
46. Topical PageRank
47. Topical HITS authority
48. Topical HITS hub
49. Inlink number
50. Outlink number
51. Number of slash in URL
52. Length of URL
53. Number of child page
54. BM25 of extracted title
55. LMIR.ABS of extracted title
56. LMIR.JM of extracted title
57. hyperlink base feature

Learning to Rank: Framework

Different approaches exist -- based on how to perform the learning (input, output, scoring functions, etc.)!
References used to prepare this set of slides:

- Nazli Goharian & Saket Mengle slides for Text Classification lecture, 2009

Other references:
- ACM Computing Surveys article by X. Qi and B. Davison, 2009
- Introduction to Information Retrieval, Manning, Raghavan and Schütze, 2008
- Introduction to Data Mining, Tan, Steinbach, Kumar, Addison Wesley, 2006
- Data Mining Concepts and Techniques, Han, Kamber, Pei, Morgan Kaufmann, 2011
- Machine Learning in Automated Text Categorization, F. Sebastiani, 2002