Learn to Weight Term in Information Retrieval Using Category Information

Rong Jin[1] Joyce Y. Chai[1] Luo Si[2]

[1]Department of Computer Science and Engineering
Michigan State University

[2]School of Computer Science
Carnegie Mellon University

International Conference on Machine Learning, 2005

Outline

1. Overview of Term Weighting Methods in Information Retrieval
   - Term Weighting based on TF.IDF
   - Term Weighting based on Language Models
   - Problems with Existing Term Weighting Methods
2. Learn Term Weights Using Category Information
   - A Framework for Learning Term Weights Using Category Information
   - A Regression Approach
   - A Probabilistic Approach
3. Experiment
   - Experimental Design
   - Baseline Approaches
   - Experimental Results
4. Summary

Outline

1. Overview of Term Weighting Methods in Information Retrieval
    - Term Weighting based on TF.IDF
    - Term Weighting based on Language Models
    - Problems with Existing Term Weighting Methods
2. Learn Term Weights Using Category Information
    - A Framework for Learning Term Weights Using Category Information
    - A Regression Approach
    - A Probabilistic Approach
3. Experiment
    - Experimental Design
    - Baseline Approaches
    - Experimental Results
4. Summary

Outline

Outline

1. Overview of Term Weighting Methods in Information Retrieval
   - Term Weighting based on TF.IDF
   - Term Weighting based on Language Models
   - Problems with Existing Term Weighting Methods
2. Learn Term Weights Using Category Information
   - A Framework for Learning Term Weights Using Category Information
   - A Regression Approach
   - A Probabilistic Approach
3. Experiment
   - Experimental Design
   - Baseline Approaches
   - Experimental Results
4. Summary

Outline

Term Weighting Methods based on TF.IDF

- Most popular methods in information retrieval.
- Consist of three factors
  - Term frequency (TF): $f(w, \mathbf{d})$
    - How frequent does the term w appear in document $\mathbf{d}$
  - Inverse document frequency (IDF):
    - How rare is term w in a collection $\mathcal{C}$

$$
\begin{aligned}
idf(w) &= \log\left(\frac{N+0.5}{N(w)}\right) \\
N &: \text{the total number of documents in collection } \mathcal{C} \\
N(w) &: \text{the number of documents in } \mathcal{C} \text{ having word } w
\end{aligned}
$$

  - Document normalization factor, e.g. $\|\mathbf{d}\|_2$
    - Reduce the bias of long documents

Term Weighting Methods based on TF.IDF

- Most popular methods in information retrieval.
- Consist of three factors
  - Term frequency (TF): $f(w, \mathbf{d})$
    - How frequent does the term w appear in document $\mathbf{d}$
  - Inverse document frequency (IDF):
    - How rare is term w in a collection $\mathcal{C}$

$$
\begin{aligned}
idf(w) &= \log\left(\frac{N+0.5}{N(w)}\right) \\
N &: \text{the total number of documents in collection } \mathcal{C} \\
N(w) &: \text{the number of documents in } \mathcal{C} \text{ having word } w
\end{aligned}
$$

  - Document normalization factor, e.g. $\|\mathbf{d}\|_2$
    - Reduce the bias of long documents

Term Weighting Methods based on TF.IDF

- Most popular methods in information retrieval.
- Consist of three factors
  - Term frequency (TF): $f(w, \mathbf{d})$
    - How frequent does the term w appear in document $\mathbf{d}$
  - Inverse document frequency (IDF):
    - How rare is term w in a collection $\mathcal{C}$

$$
\begin{aligned}
idf(w) &= \log\left(\frac{N+0.5}{N(w)}\right) \\
N &: \text{the total number of documents in collection } \mathcal{C} \\
N(w) &: \text{the number of documents in } \mathcal{C} \text{ having word } w
\end{aligned}
$$

  - Document normalization factor, e.g. $\|\mathbf{d}\|_2$
    - Reduce the bias of long documents

Term Weighting Methods based on TF.IDF

- Most popular methods in information retrieval.
- Consist of three factors
  - Term frequency (TF): $f(w, \mathbf{d})$
    - How frequent does the term w appear in document $\mathbf{d}$
  - Inverse document frequency (IDF):
    - How rare is term w in a collection $\mathcal{C}$

$$
\begin{aligned}
idf(w) &= \log\left(\frac{N+0.5}{N(w)}\right) \\
N &: \text{the total number of documents in collection } \mathcal{C} \\
N(w) &: \text{the number of documents in } \mathcal{C} \text{ having word } w
\end{aligned}
$$

  - Document normalization factor, e.g. $\|\mathbf{d}\|_2$
    - Reduce the bias of long documents

Okapi: An Example of TF.IDF Term Weighting

Similarity between query **q** and document **d** is:

$$sim(\mathbf{d}, \mathbf{q}) = \sum_{w \in \mathbf{q}} \frac{k f(w, \mathbf{q}) f(w, \mathbf{d})}{f(w, \mathbf{d}) + k(1 - b + b\frac{|\mathbf{d}|}{\overline{\mathbf{d}}})} \log\left(\frac{N + 0.5}{N(w)}\right)$$

where

$f(w, \mathbf{q})$: term frequency of w in query **q**

$f(w, \mathbf{d})$: term frequency of w in **d**

$\overline{\mathbf{d}}$: is the average document length of collection $\mathcal{C}$.

k, b : weight parameters determined empirically

Outline

Term Weighting Methods based on Language Models

- Assume each document $\mathbf{d}$ is generated by a statistical model θ_d
- Estimate θ_d by maximizing likelihood $p(\mathbf{d}|\theta_d)$
- Usually a smoothing technique, such as Jelink Mercer smoothing and Dirichlet smoothing, is used to deal with the sparse data problem

Term Weighting Methods based on Language Models

- Assume each document **d** is generated by a statistical model θ_d
- Estimate θ_d by maximizing likelihood $p(\mathbf{d}|\theta_d)$
- Usually a smoothing technique, such as Jelink Mercer smoothing and Dirichlet smoothing, is used to deal with the sparse data problem

Term Weighting Methods based on Language Models

- Assume each document **d** is generated by a statistical model θ_d
- Estimate θ_d by maximizing likelihood $p(\mathbf{d}|\theta_d)$
- Usually a smoothing technique, such as Jelink Mercer smoothing and Dirichlet smoothing, is used to deal with the sparse data problem

An Example of Language models for Information Retrieval

- The unigram language model $p(w|\mathbf{d})$ based on Jelink Mercer smoothing:

$$
\begin{aligned}
p(w|\mathbf{d}) &= (1-\alpha)p(w|\mathcal{C}) + \alpha\frac{f(w,\mathbf{d})}{|\mathbf{d}|} \\
&= p(w|\mathcal{C})\left(1-\alpha+\alpha\frac{f(w,\mathbf{d})}{|\mathbf{d}|p(w|\mathcal{C})}\right)
\end{aligned}
$$

where α is a smoothing parameter.

- The similarity of query $\mathbf{q}$ to document $\mathbf{d}$ is estimated as

$$
sim(\mathbf{q},\mathbf{d}) \propto p(\mathbf{q}|\mathbf{d}) \propto \prod_{w\in\mathbf{q}} [p(w|\mathbf{d})]^{f(w,\mathbf{q})}
$$

An Example of Language models for Information Retrieval

- The unigram language model $p(w|\mathbf{d})$ based on Jelink Mercer smoothing:

$$\begin{aligned} p(w|\mathbf{d}) &= (1-\alpha)p(w|\mathcal{C}) + \alpha\frac{f(w,\mathbf{d})}{|\mathbf{d}|} \\ &= p(w|\mathcal{C})\left(1-\alpha+\alpha\frac{f(w,\mathbf{d})}{|\mathbf{d}|p(w|\mathcal{C})}\right) \end{aligned}$$

  where α is a smoothing parameter.
- The similarity of query $\mathbf{q}$ to document $\mathbf{d}$ is estimated as

$$sim(\mathbf{q},\mathbf{d}) \propto p(\mathbf{q}|\mathbf{d}) \propto \prod_{w\in\mathbf{q}} [p(w|\mathbf{d})]^{f(w,\mathbf{q})}$$

Outline

Problems with Existing Term Weighting Methods

The essential difficulty with determining term weights is the lack of supervision.

- **Problems with TF.IDF methods**
  Either IDF or TF is sufficient to determine if a word is informative.
  - IDF factor → rare words are informative words
  - But, typos are usually rare and uninformative.
- **Problems with language modeling approaches**
  They are generative models →
  Insufficient to distinguish informative words from uninformative ones

Problems with Existing Term Weighting Methods

The essential difficulty with determining term weights is the lack of supervision.

- **Problems with TF.IDF methods**
  Either IDF or TF is sufficient to determine if a word is informative.
  - IDF factor → rare words are informative words
  - But, typos are usually rare and uninformative.
- **Problems with language modeling approaches**
  They are generative models →
  Insufficient to distinguish informative words from uninformative ones

Problems with Existing Term Weighting Methods

The essential difficulty with determining term weights is the lack of supervision.

- **Problems with TF.IDF methods**
  Either IDF or TF is sufficient to determine if a word is informative.
  - IDF factor $\rightarrow$ rare words are informative words
  - But, typos are usually rare and uninformative.
- **Problems with language modeling approaches**
  They are generative models $\rightarrow$
  Insufficient to distinguish informative words from uninformative ones

Problems with Existing Term Weighting Methods

The essential difficulty with determining term weights is the lack of supervision.

- **Problems with TF.IDF methods**
  Either IDF or TF is sufficient to determine if a word is informative.
  - IDF factor → rare words are informative words
  - But, typos are usually rare and uninformative.
- **Problems with language modeling approaches**
  They are generative models →
  Insufficient to distinguish informative words from uninformative ones

Outline

Learn Term Weights Using Category Information

- Given: each document is assigned to a set of categories
- Goal: learn term weights from the assigned categories of documents
- Main idea:
  - Each document is represented by both a bag of words and a set of categories
  - Compute document similarity based on word $s_w(\mathbf{d}_i, \mathbf{d}_j)$
  - Compute document similarity based on category $s_c(\mathbf{d}_i, \mathbf{d}_j)$
  - Find term weights $\rightarrow s_w(\mathbf{d}_i, \mathbf{d}_j) \approx s_c(\mathbf{d}_i, \mathbf{d}_j)$

Learn Term Weights Using Category Information

- Given: each document is assigned to a set of categories
- Goal: learn term weights from the assigned categories of documents
- Main idea:
  - Each document is represented by both a bag of words and a set of categories
  - Compute document similarity based on word $s_w(\mathbf{d}_i, \mathbf{d}_j)$
  - Compute document similarity based on category $s_c(\mathbf{d}_i, \mathbf{d}_j)$
  - Find term weights $\longrightarrow s_w(\mathbf{d}_i, \mathbf{d}_j) \approx s_c(\mathbf{d}_i, \mathbf{d}_j)$

Learn Term Weights Using Category Information

- Given: each document is assigned to a set of categories
- Goal: learn term weights from the assigned categories of documents
- Main idea:
  - Each document is represented by both a bag of words and a set of categories
  - Compute document similarity based on word $s_w(\mathbf{d}_i, \mathbf{d}_j)$
  - Compute document similarity based on category $s_c(\mathbf{d}_i, \mathbf{d}_j)$
  - Find term weights $\longrightarrow s_w(\mathbf{d}_i, \mathbf{d}_j) \approx s_c(\mathbf{d}_i, \mathbf{d}_j)$

Learn Term Weights Using Category Information

- Given: each document is assigned to a set of categories
- Goal: learn term weights from the assigned categories of documents
- Main idea:
  - Each document is represented by both a bag of words and a set of categories
  - Compute document similarity based on word $s_w(\mathbf{d}_i, \mathbf{d}_j)$
  - Compute document similarity based on category $s_c(\mathbf{d}_i, \mathbf{d}_j)$
  - Find term weights $\rightarrow s_w(\mathbf{d}_i, \mathbf{d}_j) \approx s_c(\mathbf{d}_i, \mathbf{d}_j)$

A Framework for Learning Term Weights Using Category Information

- For each document $\mathbf{d}_i$, we have

$$\begin{aligned} \text{Word based Rep.} \quad \mathbf{w}_i &= (w_{i,1}, w_{i,2}, ..., w_{i,n})^T \\ \text{Category based Rep.} \quad \mathbf{c}_i &= (c_{i,1}, c_{i,2}, ..., c_{i,n})^T \end{aligned}$$

- Word based document similarity

$$s_w(\mathbf{d}_i, \mathbf{d}_j; \mu) = \sum_{k=1}^{m} \mu_k w_{i,k} w_{j,k}$$

- Category based document similarity

$$s_c(\mathbf{d}_i, \mathbf{d}_j; \eta) = \sum_{k=1}^{m} \eta_k c_{i,k} c_{j,k}$$

A Framework for Learning Term Weights Using Category Information

- For each document $\mathbf{d}_i$, we have

$$\begin{aligned} \text{Word based Rep.} \quad \mathbf{w}_i &= (w_{i,1}, w_{i,2}, ..., w_{i,n})^T \\ \text{Category based Rep.} \quad \mathbf{c}_i &= (c_{i,1}, c_{i,2}, ..., c_{i,n})^T \end{aligned}$$

- Word based document similarity

$$s_w(\mathbf{d}_i, \mathbf{d}_j; \mu) = \sum_{k=1}^{m} \mu_k w_{i,k} w_{j,k}$$

- Category based document similarity

$$s_c(\mathbf{d}_i, \mathbf{d}_j; \eta) = \sum_{k=1}^{m} \eta_k c_{i,k} c_{j,k}$$

A Framework for Learning Term Weights Using Category Information (Cont'd)

- Find weights η and μ s.t. $s_w(\mathbf{d}_i, \mathbf{d}_j; \mu) \approx s_c(\mathbf{d}_i, \mathbf{d}_j; \eta)$ for any two documents $\mathbf{d}_i$ and $\mathbf{d}_j$

$$(\eta^*, \mu^*) = \arg\min_{\eta,\mu} \sum_{i \neq j} l(s_c(\mathbf{d}_i, \mathbf{d}_j; \eta), s_w(\mathbf{d}_i, \mathbf{d}_j; \mu))$$

where $l(x, y)$ is a loss function measures the difference between x and y.

Outline

A Regression Approach Toward Learning Term Weights

- Define loss function $l(s_c, s_w) = \|s_c - s_w\|^2$
- Objective function $\mathcal{F}_{reg}$

$$\mathcal{F}_{reg} = (\eta^T, \mu^T) \begin{pmatrix} Q_c & -P^T \\ -P & Q_w \end{pmatrix} \begin{pmatrix} \eta \\ \mu \end{pmatrix}$$

where

$$[Q_w]_{i,j} = (\mathbf{u}_i^T \mathbf{u}_j)^2, [Q_c]_{i,j} = (\mathbf{v}_i^T \mathbf{v})^2, [P]_{i,j} = (\mathbf{u}_i^T \mathbf{v}_j)^2$$

$\mathbf{u}_i$: frequency vector for the i-th term

$\mathbf{v}_i$: frequency vector for the j-th category

A Regression Approach Toward Learning Term Weights

- Define loss function $l(s_c, s_w) = \|s_c - s_w\|^2$
- Objective function $\mathcal{F}_{reg}$

$$\mathcal{F}_{reg} = (\eta^T, \mu^T) \begin{pmatrix} Q_c & -P^T \\ -P & Q_w \end{pmatrix} \begin{pmatrix} \eta \\ \mu \end{pmatrix}$$

where

$$[Q_w]_{i,j} = (\mathbf{u}_i^T \mathbf{u}_j)^2, [Q_c]_{i,j} = (\mathbf{v}_i^T \mathbf{v})^2, [P]_{i,j} = (\mathbf{u}_i^T \mathbf{v}_j)^2$$

$\mathbf{u}_i$: frequency vector for the i-th term

$\mathbf{v}_i$: frequency vector for the j-th category

The Regression Approach: Constraints

- Trivial solution $\eta = \mu = 0 \rightarrow \mathcal{F}_{reg} = 0$
- L2 Constraint:

$$\|\eta\|_2^2 + \|\mu\|_2^2 \geq 1$$

  - Problem: negative term weight $\mu_i < 0$
    $\rightarrow$ When two documents share word w_i, they are less likely to be similar
- L1 Constraint:

$$\eta_i \geq 0; \quad \mu_j \geq 0$$

$$\sum_{i=1}^{m} \eta_i + \sum_{i=1}^{n} \mu_i \geq 1$$

The Regression Approach: Constraints

- Trivial solution $\eta = \mu = 0 \rightarrow \mathcal{F}_{reg} = 0$
- L2 Constraint:

$$\|\eta\|_2^2 + \|\mu\|_2^2 \geq 1$$

  - Problem: negative term weight $\mu_i < 0$
    $\rightarrow$ When two documents share word w_i, they are less likely to be similar
- L1 Constraint:

$$\eta_i \geq 0; \quad \mu_j \geq 0$$

$$\sum_{i=1}^{m} \eta_i + \sum_{i=1}^{n} \mu_i \geq 1$$

The Regression Approach: Constraints

- Trivial solution $\eta = \mu = 0 \rightarrow \mathcal{F}_{reg} = 0$
- L2 Constraint:

$$\|\eta\|_2^2 + \|\mu\|_2^2 \geq 1$$

  - Problem: negative term weight $\mu_i < 0$
    $\rightarrow$ When two documents share word w_i, they are less likely to be similar
- L1 Constraint:

$$\eta_i \geq 0; \quad \mu_j \geq 0$$

$$\sum_{i=1}^{m} \eta_i + \sum_{i=1}^{n} \mu_i \geq 1$$

The Regression Approach: Constraints

- Trivial solution $\eta = \mu = 0 \rightarrow \mathcal{F}_{reg} = 0$
- L2 Constraint:

$$\|\eta\|_2^2 + \|\mu\|_2^2 \geq 1$$

  - Problem: negative term weight $\mu_i < 0$
    $\rightarrow$ When two documents share word w_i, they are less likely to be similar
- L1 Constraint:

$$\eta_i \geq 0; \quad \mu_j \geq 0$$
$$\sum_{i=1}^{m} \eta_i + \sum_{i=1}^{n} \mu_i \geq 1$$

The Regression Approach: Final Form

- Final form for the regression approach

$$\begin{aligned} \min_{\eta,\mu} \quad & (\eta^T, \mu^T) \begin{pmatrix} Q_c & -P^T \\ -P & Q_w \end{pmatrix} \begin{pmatrix} \eta \\ \mu \end{pmatrix} \\ \text{s. t} \quad & \eta \succeq \mathbf{0}, \ \mu \succeq \mathbf{0} \\ & \|\eta\|_1 + \|\mu\|_1 \geq 1 \end{aligned}$$

- Solve by quadratic programming techiques

Outline

A Probabilistic Approach Toward Learning Term Weights

- Probability for documents to be similar based on words

$$p_{i,j}^{w} = \frac{1}{1 + \exp\left(-s_w(\mathbf{d}_i, \mathbf{d}_j; \mu) + \mu_0\right)}$$

- Probability for documents to be similar based on categories

$$p_{i,j}^{c} = \frac{1}{1 + \exp\left(-s_c(\mathbf{d}_i, \mathbf{d}_j; \eta) + \eta_0\right)}$$

- Loss function: cross entropy function

$$l(s_c(\mathbf{d}_i, \mathbf{d}_j; \eta), s_w(\mathbf{d}_i, \mathbf{d}_j; \mu)) = -p_{i,j}^{c} \log p_{i,j}^{w} - (1 - p_{i,j}^{c}) \log(1 - p_{i,j}^{w})$$

A Probabilistic Approach Toward Learning Term Weights

- Probability for documents to be similar based on words

$$p^w_{i,j} = \frac{1}{1 + \exp\left(-s_w(\mathbf{d}_i, \mathbf{d}_j; \mu) + \mu_0\right)}$$

- Probability for documents to be similar based on categories

$$p^c_{i,j} = \frac{1}{1 + \exp\left(-s_c(\mathbf{d}_i, \mathbf{d}_j; \eta) + \eta_0\right)}$$

- Loss function: cross entropy function

$$l(s_c(\mathbf{d}_i, \mathbf{d}_j; \eta), s_w(\mathbf{d}_i, \mathbf{d}_j; \mu)) = \\ -p^c_{i,j} \log p^w_{i,j} - (1 - p^c_{i,j}) \log(1 - p^w_{i,j})$$

The Probabilistic Approach: Final Form

- Objective function $\mathcal{F}_{prob}$

$$\mathcal{F}_{prob} = \sum_{i \neq j}^{N} p^c_{i,j} \log p^w_{i,j} + (1 - p^c_{i,j}) \log(1 - p^w_{i,j})$$

- The final form for the probabilistic approach:

$$\begin{array}{ll} \arg \max\limits_{\eta, \mu, \eta_0, \mu_0} & \mathcal{F}_{prob} - \alpha_w \sum_{i=1}^{n} \mu_i - \alpha_c \sum_{i=1}^{m} \eta_i \\ \text{s. t.} & \eta \succeq 0, \ \mu \succeq 0 \end{array}$$

where $\alpha_w > 0$ and $\alpha_c > 0$ are regularization parameters.

The Probabilistic Approach: Optimization Strategy

Alternating Optimization

- Learn term weights μ with fixed category weights η
  - Decouple the correlation among μ

$$\mathcal{F}_{prob}(\mu', \eta) - \mathcal{F}_{prob}(\mu, \eta) \geq \sum_{i=1}^{n} g_i(\mu'_i - \mu_i)$$

  μ' and μ are term weights of two consecutive iterations.
  - Solve

$$g'_i(\delta_i) = 0 \rightarrow \mu' = \mu + \delta$$

- Learn category weights η with fixed term weights μ
  - A similar procedure for optimizing η with fixed μ

The Probabilistic Approach: Optimization Strategy

Alternating Optimization

- Learn term weights μ with fixed category weights η
  - Decouple the correlation among μ

$$\mathcal{F}_{prob}(\mu', \eta) - \mathcal{F}_{prob}(\mu, \eta) \geq \sum_{i=1}^{n} g_i(\mu'_i - \mu_i)$$

  μ' and μ are term weights of two consecutive iterations.
  - Solve

$$g'_i(\delta_i) = 0 \rightarrow \mu' = \mu + \delta$$

- Learn category weights η with fixed term weights μ
  - A similar procedure for optimizing η with fixed μ

The Probabilistic Approach: Optimization Strategy

Alternating Optimization

- Learn term weights μ with fixed category weights η
  - Decouple the correlation among μ

$$\mathcal{F}_{prob}(\mu', \eta) - \mathcal{F}_{prob}(\mu, \eta) \geq \sum_{i=1}^{n} g_i(\mu'_i - \mu_i)$$

  μ' and μ are term weights of two consecutive iterations.
  - Solve

$$g'_i(\delta_i) = 0 \rightarrow \mu' = \mu + \delta$$

- Learn category weights η with fixed term weights μ
  - A similar procedure for optimizing η with fixed μ

The Probabilistic Approach: Optimization Strategy

Alternating Optimization

- Learn term weights μ with fixed category weights η
  - Decouple the correlation among μ

$$\mathcal{F}_{prob}(\mu', \eta) - \mathcal{F}_{prob}(\mu, \eta) \geq \sum_{i=1}^{n} g_i(\mu'_i - \mu_i)$$

  μ' and μ are term weights of two consecutive iterations.
  - Solve

$$g'_i(\delta_i) = 0 \rightarrow \mu' = \mu + \delta$$

- Learn category weights η with fixed term weights μ
  - A similar procedure for optimizing η with fixed μ

Outline

Experimental Design

- Document collection
  - A document collection from the ad hoc retrieval task of ImageCLEF
  - Totally 28,133 documents, 933 categories
  - Average document length ~ 50
  - Average number of categories for a document ~ 5
- Evaluation Queries
  - 5 queries from ImageCLEF 2003 for training α_w and α_c
  - 25 queries from ImageCLEF 2004 for testing
- Evaluation metrics
  - Average precision for top retried documents
  - Average precision across 11 recall points
  - Precision recall curve

Experimental Design

- Document collection
  - A document collection from the ad hoc retrieval task of ImageCLEF
  - Totally 28,133 documents, 933 categories
  - Average document length ~ 50
  - Average number of categories for a document ~ 5
- Evaluation Queries
  - 5 queries from ImageCLEF 2003 for training α_w and α_c
  - 25 queries from ImageCLEF 2004 for testing
- Evaluation metrics
  - Average precision for top retried documents
  - Average precision across 11 recall points
  - Precision recall curve

Experimental Design

- Document collection
  - A document collection from the ad hoc retrieval task of ImageCLEF
  - Totally 28,133 documents, 933 categories
  - Average document length ~ 50
  - Average number of categories for a document ~ 5
- Evaluation Queries
  - 5 queries from ImageCLEF 2003 for training α_w and α_c
  - 25 queries from ImageCLEF 2004 for testing
- Evaluation metrics
  - Average precision for top retried documents
  - Average precision across 11 recall points
  - Precision recall curve

Outline

Baseline Approaches

- State-of-art information retrieval methods
  - The Okapi method (**Okapi**)
  - The language model with JM smoothing (**LM**)
- Inverse category frequency (**ICF**)

$$icf(w) = \log\left(\frac{m}{m(w)}\right)$$
$$m(w) : \text{number of categories having word } w$$

  - Replace $idf(w)$ with $icf(w)$ in the Okapi method

Baseline Approaches (Cont'd)

- Category-based query expansion (**CQE**)
  1. Retrieve top $k = 100$ documents for query $\mathbf{q}$ using Okapi
  2. Expand query $\mathbf{q}$ to include category information

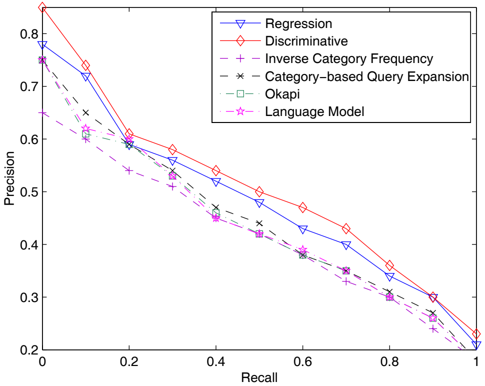
$$\begin{aligned} \mathbf{q}' &= \{f(w_1, \mathbf{q}), ..., f(w_n, \mathbf{q}); f(c_1, \mathbf{q}), ..., f(c_m, \mathbf{q})\} \\ f(c_i, \mathbf{q}) &: \text{the number of top } k \text{ documents in category } c_i \end{aligned}$$

  3. Retrieve documents using the expanded query $\mathbf{q}'$

$$\log p(\mathbf{q}'|\mathbf{d}) = \frac{\beta \sum_{i=1}^{n} f(w_i, \mathbf{q}) \log p(w_i|\mathbf{d})}{\sum_{i=1}^{n} f(w_i, \mathbf{q})} + \frac{(1-\beta) \sum_{i=1}^{m} f(c_i, \mathbf{q}) \log p(c_i|\mathbf{d})}{\sum_{i=1}^{m} f(c_i, \mathbf{q})}$$

Outline

Precision Recall Curves



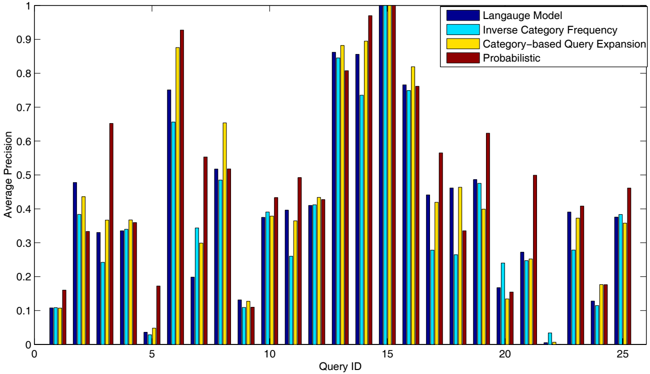
- Probabilistic approach > Language Model & Okapi

Average Precision

	Using Category				No Category	
	Reg.	Prob.	ICF	CQE	Okapi	LM
Avg. Prec.	0.45	**0.48**	0.38	0.42	0.41	0.41
Prec @ 5 doc	0.55	**0.56**	0.40	0.50	0.47	0.50
Prec @ 10 doc	0.48	**0.52**	0.40	0.48	0.45	0.48
Prec @ 20 doc	**0.46**	**0.46**	0.39	0.42	0.39	0.38
Prec @ 100 doc	**0.21**	**0.21**	0.19	0.19	0.20	0.20

- Reg. and Prob. > Okapi and LM
  - Category information is useful
- ICF and CQE < Okapi and LM
  - Need to exploit category information wisely

Retrieval Precision for Individual Queries



- Over 16 queries, probabilistic approach > langauge model
- Over 5 queries, probabilistic approach < langauge model

Summary

- Proposed two algorithms for learning term weights using category information
  - A regression approach
  - A probabilistic approach
- Empirical studies with the ImageCLEF dataset verify the effectiveness of the proposed algorithms
- Future work
  - Improve learning efficiency for large numbers of documents and large-sized vocabularies
  - Extend to image retrieval for annotated images