



Integrating Term Dependencies in IR

Jian-Yun Nie
RALI, Dept. IRO
Université de Montréal



Overview

- Traditional models
 - Assumption of term independence
 - A term is different from another in meaning (last lecture)
 - A term has a unique meaning (this lecture)
 - Other implicit assumptions
 - The meaning of a term is independent from its context
- Possible solutions
 - Extend unigram model to bi-/tri-gram models
 - Use phrases
 - Use linguistic/statistical dependencies
 - Use term proximity
- Question
 - What problems remain?



Recall: n-grams

- Uni-gram:
$$P(s) = \prod_{i=1}^n P(w_i)$$
- Bi-gram:
$$P(s) = \prod_{i=1}^n P(w_i | w_{i-1})$$
- Tri-gram:
$$P(s) = \prod_{i=1}^n P(w_i | w_{i-2}w_{i-1})$$



Beyond uni-grams

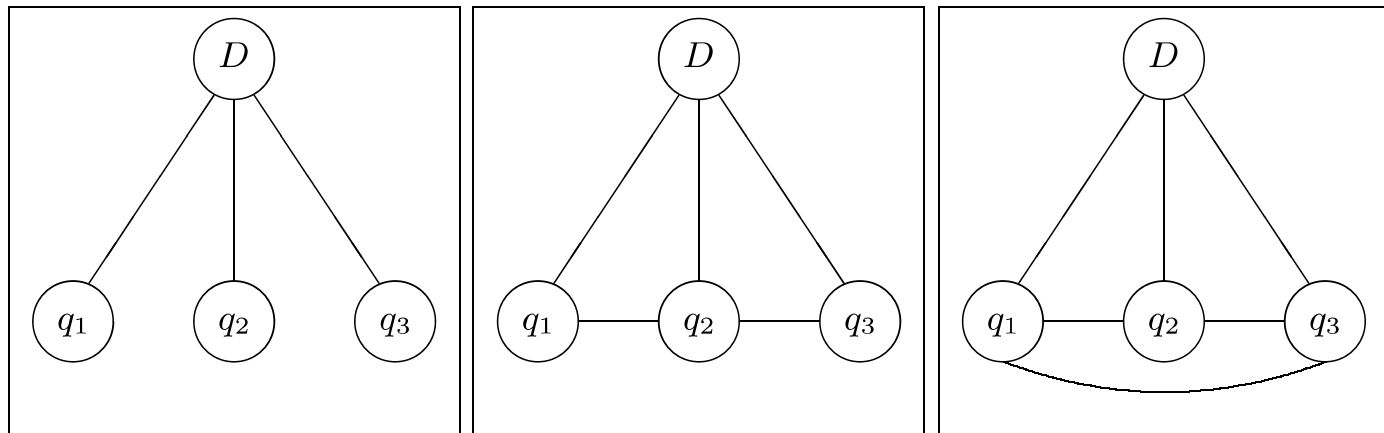
- Using Bi-grams [Song and Croft, 99]

$$P(w_i | w_{i-1}, D) = \lambda_1 P_{MLE}(w_i | w_{i-1} D) + \lambda_2 P_{MLE}(w_i | D) + \lambda_3 P_{MLE}(w_i | C)$$

- Bi-term [Srikanth and Srihari, 02]
 - Do not consider word order in bi-grams
(analysis, data) – (data, analysis)
- Results:
 - Bi-gram model is slightly better than unigram model, but much more expensive
 - Bi-term model is slightly better than bi-gram model

Markov Random Field Model

- Consider connections between query terms



no connection

sequential

full

- Model cliques



MRF - model

- Joint probability:

$$P_{\Lambda}(Q, D) = \frac{1}{Z_{\Lambda}} \prod_{c \in C(G)} \psi(c; \Lambda)$$

$$Q = q_1 \dots q_n$$

$C(G)$ is the set of cliques in G

$\psi(\cdot; \Lambda)$ is a non-negative *potential function*

$$Z_{\Lambda} = \sum_{Q, D} \prod_{c \in C(G)} \psi(c; \Lambda)$$



MRF - model

- Ranking function:

$$P_{\Lambda}(D|Q) = \frac{P_{\Lambda}(Q, D)}{P_{\Lambda}(Q)}$$

$$\stackrel{\text{rank}}{=} \log P_{\Lambda}(Q, D) - \log P_{\Lambda}(Q)$$

$$\stackrel{\text{rank}}{=} \sum_{c \in C(G)} \log \psi(c; \Lambda)$$

$$\psi(c; \Lambda) = \exp[\lambda_c f(c)]$$

$f(c)$ is some real-valued *feature function*

λ_c is the weight

$$P_{\Lambda}(D|Q) \stackrel{\text{rank}}{=} \sum_{c \in C(G)} \lambda_c f(c)$$



MRF - features

- Single term

$$\begin{aligned}\psi_T(c) &= \lambda_T \log P(q_i|D) \\ &= \lambda_T \log \left[(1 - \alpha_D) \frac{tf_{q_i,D}}{|D|} + \alpha_D \frac{cf_{q_i}}{|C|} \right]\end{aligned}$$

- Group of terms

$$\begin{aligned}\psi_O(c) &= \lambda_O \log P(\#1(q_i, \dots, q_{i+k})|D) \\ &= \lambda_O \log \left[(1 - \alpha_D) \frac{tf_{\#1(q_i \dots q_{i+k}),D}}{|D|} + \alpha_D \frac{cf_{\#1(q_i \dots q_{i+k})}}{|C|} \right]\end{aligned}$$

Exact phrase

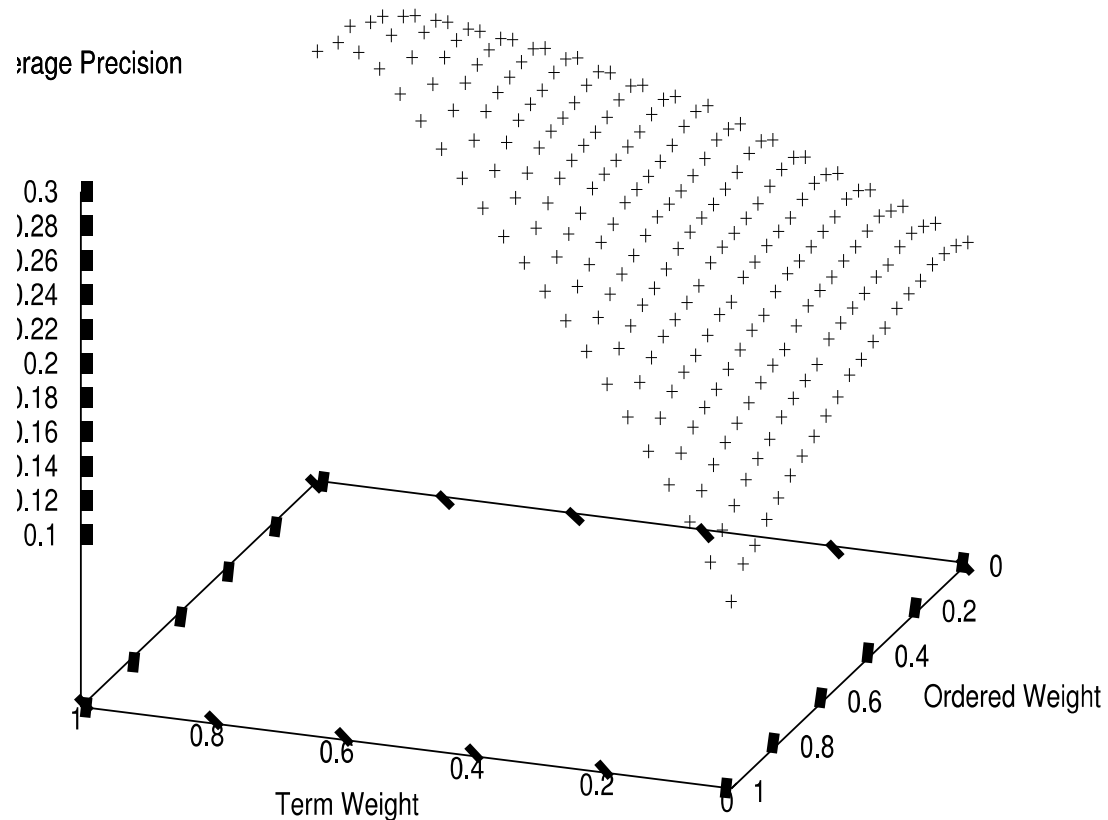
$$\begin{aligned}\psi_U(c) &= \lambda_U \log P(\#uwN(q_i, \dots, q_j)|D) \\ &= \lambda_U \log \left[(1 - \alpha_D) \frac{tf_{\#uwN(q_i \dots q_j),D}}{|D|} + \alpha_D \frac{cf_{\#uwN(q_i \dots q_j)}}{|C|} \right]\end{aligned}$$

Within window

MRF - parameters

- Set parameters to maximize MAP on training collection

$$\lambda_T + \lambda_O + \lambda_U = 1$$





MRF - results

- Unigram

| | AP | WSJ | WT10g | GOV2 |
|--------|--------|--------|--------|--------|
| AvgP | 0.1775 | 0.2592 | 0.2032 | 0.2502 |
| P @ 10 | 0.2912 | 0.4327 | 0.2866 | 0.4837 |
| μ | 3000 | 3500 | 4000 | 4000 |

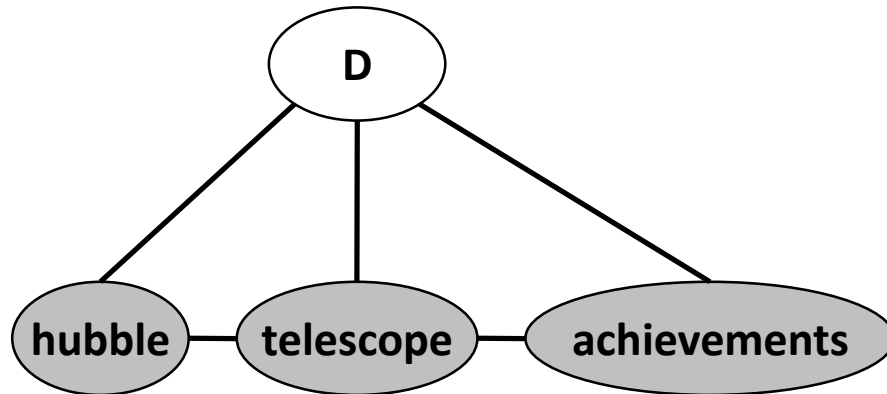
- Sequential (different window sizes)

| Length | AP | WSJ | WT10g | GOV2 |
|-----------|--------|--------|--------|--------|
| 2 | 0.1860 | 0.2776 | 0.2148 | 0.2697 |
| 8 | 0.1867 | 0.2763 | 0.2167 | 0.2832 |
| 50 | 0.1858 | 0.2766 | 0.2154 | 0.2817 |
| Unlimited | 0.1857 | 0.2759 | 0.2138 | 0.2714 |

- Full: little change from Sequential

Discussions on MRF model

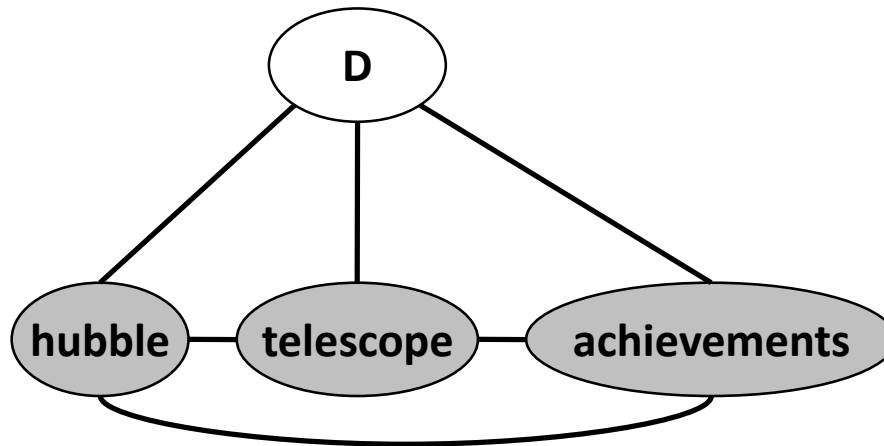
- Sequential model: consider connections between adjacent terms. Is this reasonable?



- How to extend to connections of longer distance?

Discussions on MRF model

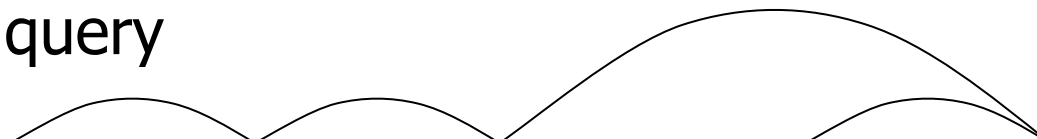
- Why isn't Full model better than Sequential model?



Beyond adjacent term dependencies

- Dependence LM (Gao et al. 04):
Consider more distant dependencies
 - Syntactic analysis
 - Statistical analysis
 - Only retain the most probable dependencies in the query

(how) (has) affirmative action affected (the) construction industry





Dependence model

- Ranking function:

$$P(Q | D) = \sum_L P(Q, L | D) = \sum_L P(L | D)P(Q | L, D)$$

- Use all the possible Linkages L (a linkage=a complete link graph)
- Approximation: Use the strongest linkage

$$P(Q | D) = P(L | D)P(Q | L, D)$$

such that $L = \arg \max_L P(L|Q)$.



Choose the strongest linkage

- Choose the connections that maximize a global measure of dependency:

$$P(L|Q) = \prod_{l \in L} P(l|Q)$$

- The connections obey some constraints:
 - acyclic and planar
 - Every term is connected
 - No cycle
 - No link crossing



Choose the strongest linkage

- Weight of one link

$$F(R | q_i, q_j) = \frac{C(q_i, q_j, R)}{C(q_i, q_j)}$$

- The best linkage:

$$L = \arg \max_L P(L | Q) = \arg \max_L \prod_{(i,j) \in L} F(R | q_i, q_j)$$

Estimate the prob. of links (EM)



For a corpus C:

1. Initialization: link each pair of words with a window of 3 words
2. For each sentence in C:
 - Apply the link prob. to select the strongest links that cover the sentence
3. Re-estimate link prob.
4. Repeat 2 and 3

Result: prob. of link $a-b$ in a language



Calculation of $P(Q|D)$

1. Determine the links in Q (the required links)

$$L = \arg \max_L P(L | Q) = \arg \max_L \prod_{(i,j) \in L} P_C(R | q_i, q_j)$$

2. Calculate the likelihood of Q (words and links)

$$P(Q | D) = P(L | D)P(Q | L, D)$$

$$P(L | D) = \prod_{l \in L} P(l | D)$$

$$\begin{aligned} P(Q | L, D) &= P(q_h | D) \prod_{(i,j) \in L} P(q_j | q_i, L, D) = \dots \\ &= \prod_{i=1..n} P(q_i | D) \prod_{(i,j) \in L} \frac{P(q_i, q_j | L, D)}{P(q_i | D)P(q_j | D)} \end{aligned}$$

$$\log P(Q | D) = \log P(L | D) + \sum_{i=1..m} \log P(q_i | D)$$

$$+ \sum_{(i,j) \in L} MI(q_i, q_j | L, D)$$



Extension to consider word relationships

- What [Gao et al. 04] tried to do:
 - Consider the constraints of words
 - Consider the constraints of linkage
- Extension of the bi-gram model
 - The pairs of terms are not always adjacent
- See experimental results in [Gao et al. 04]: Improvements



Experiments in [Gao et al. 04]

| Models | WSJ | | | PAT | | | FR | | |
|------------|--------------|------------------|------------------|--------------|-----------------|------------------|--------------|------------------|------------------|
| | AvgP | % change over BM | % change over UG | AvgP | %change over BM | % change over UG | AvgP | % change over BM | % change over UG |
| BM | 22.30 | -- | -- | 26.34 | -- | -- | 15.96 | -- | -- |
| UG | 17.91 | -19.69** | -- | 25.47 | -3.30 | -- | 14.26 | -10.65 | -- |
| DM | 22.41 | +0.49 | +25.13** | 30.74 | +16.70 | +20.69 | 17.82 | +11.65* | +24.96* |
| BG | 21.46 | -3.77 | +19.82 | 29.36 | +11.47 | +15.27 | 15.65 | -1.94 | +9.75 |
| BT1 | 21.67 | -2.83 | +20.99* | 28.91 | +9.76 | +13.51 | 15.71 | -1.57 | +10.17 |
| BT2 | 18.66 | -16.32 | +4.19 | 28.22 | +7.14 | +10.80 | 14.77 | -7.46 | +3.58 |

Table 2. Comparison results on **WSJ**, **PAT** and **FR** collections. * and ** indicate that the difference is statistically significant according to t-test (* indicates p -value < 0.05, ** indicates p -value < 0.02).

| Models | SJM | | | AP | | | ZIFF | | |
|------------|--------------|------------------|------------------|--------------|-----------------|------------------|--------------|------------------|------------------|
| | AvgP | % change over BM | % change over UG | AvgP | %change over BM | % change over UG | AvgP | % change over BM | % change over UG |
| BM | 19.14 | -- | -- | 25.34 | -- | -- | 15.36 | -- | -- |
| UG | 20.68 | +8.05 | -- | 24.58 | -3.00 | -- | 16.47 | +7.23 | -- |
| DM | 24.72 | +29.15* | +19.54** | 25.87 | +2.09 | +5.25** | 18.18 | +18.36* | +10.38** |
| BG | 24.60 | +28.53* | +18.96** | 26.24 | +3.55 | +6.75* | 17.17 | +11.78 | +4.25 |
| BT1 | 23.29 | +21.68 | +12.62** | 25.90 | +2.21 | +5.37 | 17.66 | +14.97 | +7.23 |
| BT2 | 21.62 | +12.96 | +4.55 | 25.43 | +0.36 | +3.46 | 16.34 | +6.38 | -0.79 |

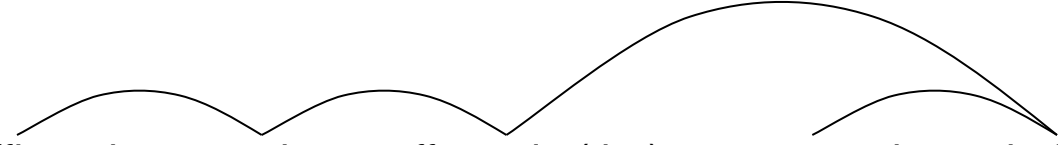
Table 3. Comparison results on **SJM**, **AP** and **ZIFF** collections. * and ** indicate that the difference is statistically significant according to t-test (* indicates p -value < 0.05, ** indicates p -value < 0.02).



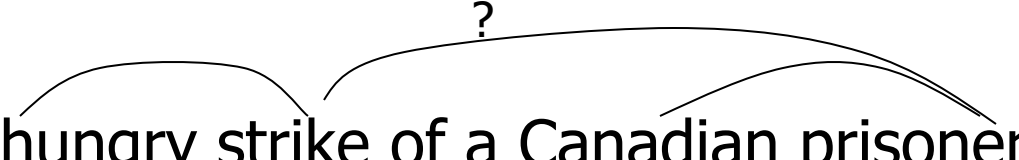
Problem

- Each word has to be connected to another word

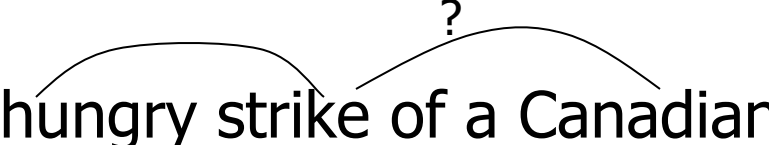
(how) (has) affirmative action affected (the) construction industry



hungry strike of a Canadian prisoner



hungry strike of a Canadian





Possible solution-1

- Keep the dependencies that are strong enough
- Alternative: consider dependencies only within compound terms

hungry strike of a Canadian

- Recognize compound terms
- Estimate dependencies



Using nouns phrases

- Detect noun phrases
 - Using a phrase dictionary
 - Using an NLP analysis (e.g. shallow parsing)
- Combining a phrase model with a word model

$$\text{score}(Q, D) = \lambda \text{score}_{\text{phrase}}(Q, D) + (1 - \lambda) \text{score}_{\text{word}}(Q, D)$$

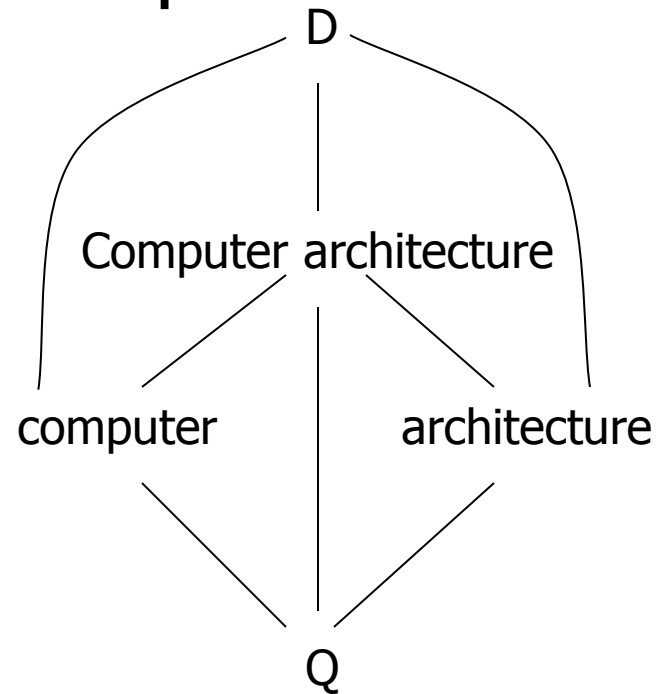
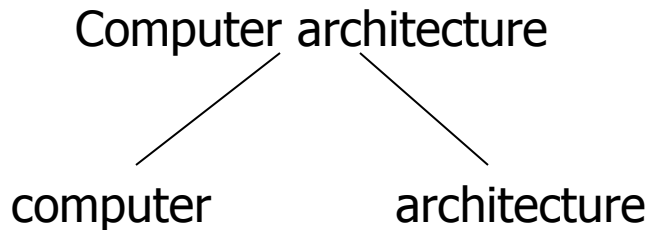
$$\text{score}_{\text{phrase}}(Q, D) = \sum_{C \in \hat{C}(Q)} \log P(C | D)$$

$$\text{score}_{\text{word}}(Q, D) = \sum_{q \in \hat{q}(Q)} \log P(q | D)$$

- The idea can also be used in other models (vector space model, ...)

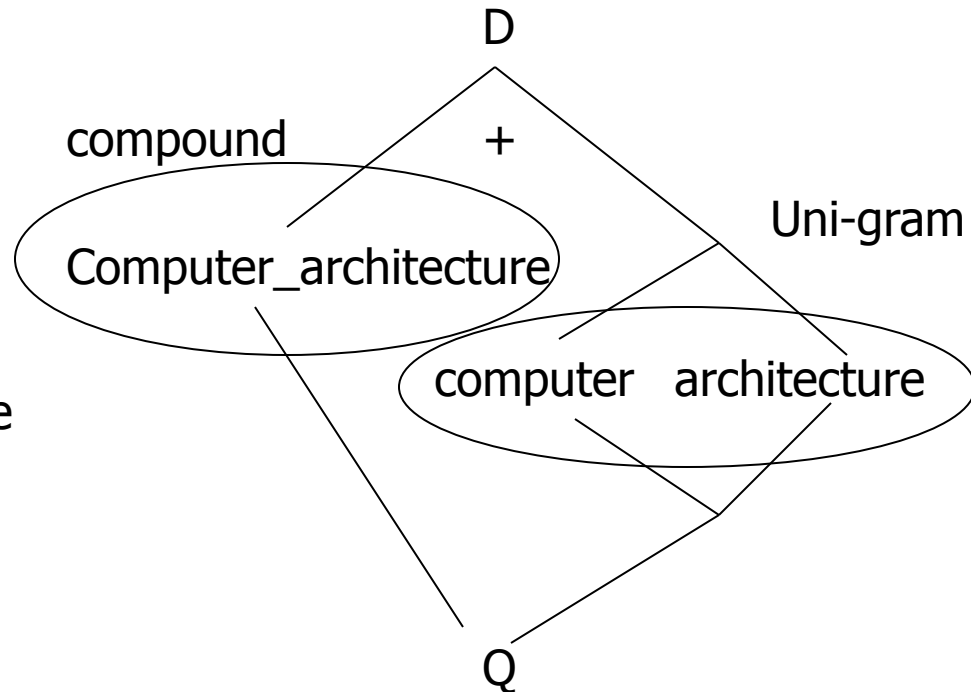
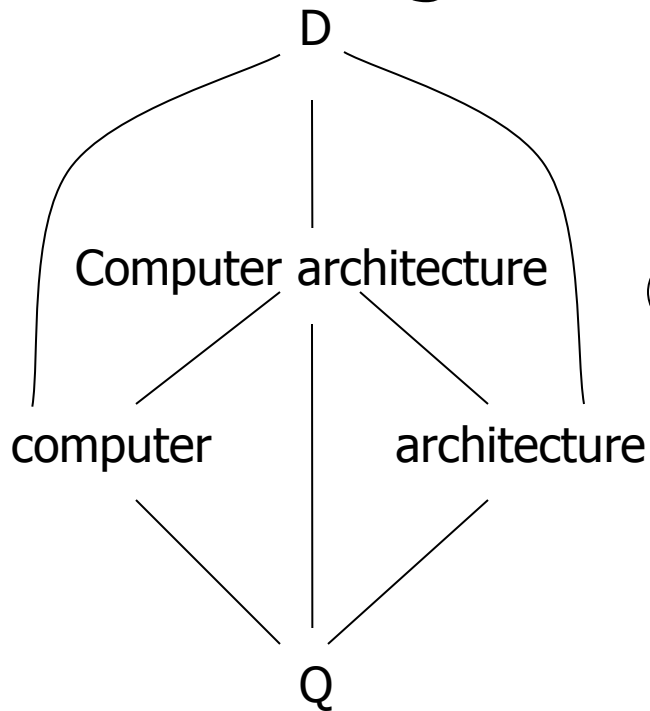
Possible solution-2

- Create a structured representation for a text: single words, compounds



Possible solution-2

- Compound model + uni-gram model = smoothing





Possible solution-2

$$\begin{aligned} &P(\textit{computer architecture} | D) \\ &= \lambda P(\textit{computer_architecture} | D) + \\ &\quad (1 - \lambda) P(\textit{computer, architecture} | D) \end{aligned}$$

- But what λ to set?
- Should it be dependent on the phrase?
- Is this solution problematic?



A partial solution

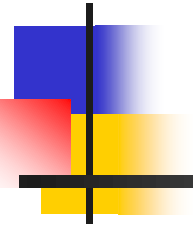
- Allow different dependencies to have different weights
 - Weight = how useful the dependency is for IR
- Presentation of a paper of [Shi and Nie, AIRS'10]



Term proximity – a flexible dependence

- Assumption commonly used in search engines:
 - A document in which query terms appear closely is preferred
 - If the terms appear in order, it is preferred
- Attempts in IR research to confirm the assumption

An Exploration of Proximity Measures in Information Retrieval



Tao Tao, Microsoft Corp.
ChengXiang Zhai, University of
Illinois at Urbana-Champaign

Published in SIGIR'07

Motivation

Heuristics to measure proximity

Query: <space program>

Document 1

.....
however, the first practical solar cell was not introduced until 1900 in response to the **program of the space**, this first solar photovoltaic cell were made of single crystal silicon and show about 50 percent efficiency

Document 2

.....
film have been determine in from **space** charge limit current measure.
.....
this paper summarizes the result of a **program** initial at the naval research laboratory

.....
Document 1 is more relevant than document 2, since the two query words are closer to each other.

Measuring proximity :Five heuristics:

- 1. **[Span]**

Query: <t1,t2>

t1, t2, t1, t3 , t5 , t4, t2, t3, t4



Span = 7

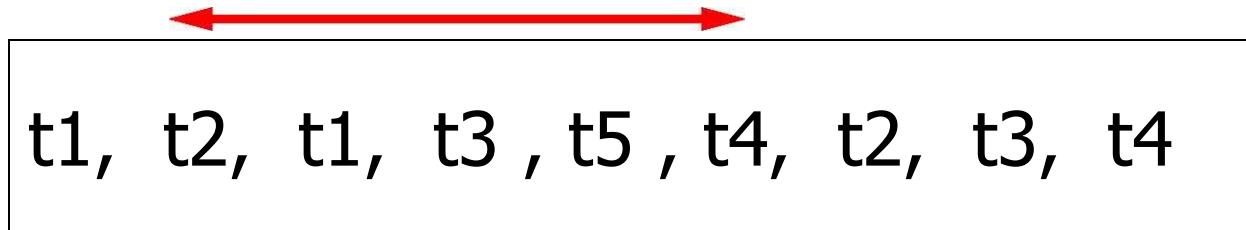
[Span]:The length of the segment from **the first query word to the last query word.**



Five heuristics:

- 2、 [MinCover]

Query: $\langle t1, t2, t4 \rangle$

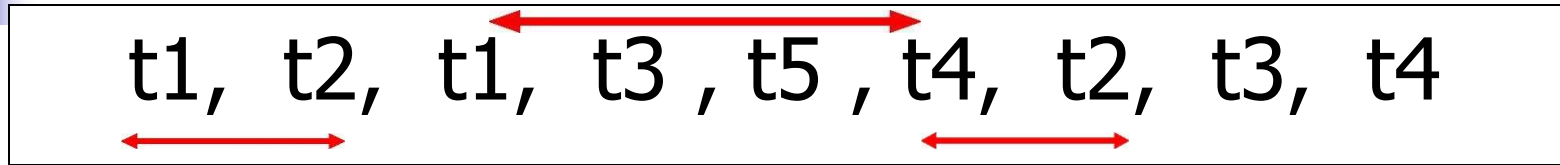


MinCover = 5

[MinCover]:The length of the minimum segment to cover all query words **at least once**.

Five heuristics: pair-wise distances

Query: $\langle t1, t2, t4 \rangle$



$\langle t1, t2 \rangle$

$\langle t1, t4 \rangle$

$\langle t2, t4 \rangle$

distance = 2

distance = 4

distance = 2

aggregation

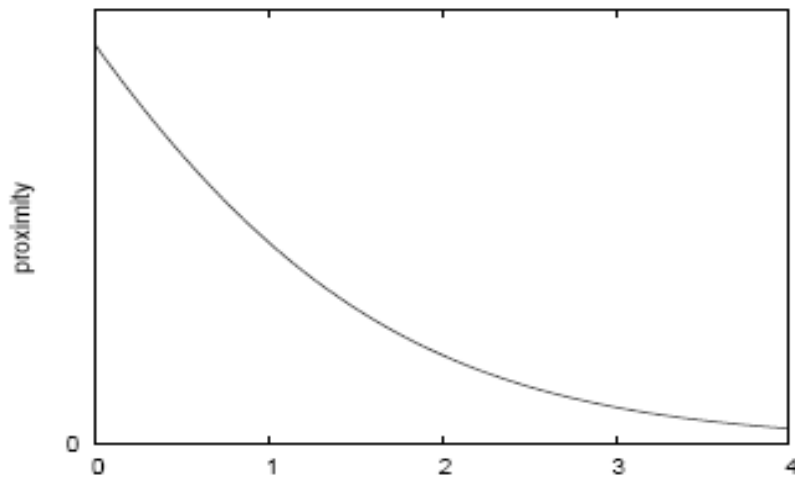
MinDist
= 2

AveDist
= $8/3$

MaxDist
= 4

Proximity retrieval models – distance-based score

- 1) The smaller the distance is, the larger the relevance is
- 2) Drop quickly in the beginning, and go flat in the end



$$\pi(Q, D) = \log(\alpha + \exp(-\delta(Q, D)))$$

Distance measure



Proximity retrieval models

- Incorporating proximities into other retrieval models

$$R_1(Q, D) = KL(Q, D) + \pi(Q, D)$$

$$R_2(Q, D) = BM25(Q, D) + \pi(Q, D)$$



Experiment

| | method/data | AP | DOE | FR | TREC8 | WEB2g |
|-------|-------------|----------------|----------------|---------------|----------------|----------------|
| R_1 | KL | 0.2220 | 0.1803 | 0.2442 | 0.2509 | 0.3008 |
| | Span | 0.2203 | 0.1717 | 0.2436 | 0.2511 | 0.2992 |
| | MinCover | 0.2200 | 0.1685 | 0.2659 | 0.2455 | 0.2947 |
| | MinDist | 0.2265* | 0.2018* | 0.2718 | 0.2573* | 0.3276* |
| | AveDist | 0.2244 | 0.1922 | 0.2683 | 0.2538 | 0.3079 |
| | MaxDist | 0.2247 | 0.1913 | 0.2687 | 0.2536 | 0.2966 |
| R_2 | BM25 | 0.2302 | 0.1840 | 0.3089 | 0.2512 | 0.3094 |
| | Span | 0.2292 | 0.1808 | 0.3101 | 0.2468 | 0.3073 |
| | MinCover | 0.2260 | 0.1815 | 0.2881 | 0.2260 | 0.2966 |
| | MinDist | 0.2368* | 0.2023* | 0.3135 | 0.2585* | 0.3395* |
| | AveDist | 0.2314 | 0.1960 | 0.3115 | 0.2506 | 0.3148 |
| | MaxDist | 0.2323 | 0.1942 | 0.3115 | 0.2492 | 0.3144 |



Positional model [Lv and Zhai, 09]

- Idea

- A model at each position within a document:
 $P(w|D,i)$
- Count at a position is propagated to the positions around it according to some functions
- Score of a document at position i

$$S(Q,D,i) = - \sum_{w \in V} p(w|Q) \log \frac{p(w|Q)}{p(w|D,i)}$$

- Document score = combining scores at different positions



Positional LM

- $c(w, i)$: the count of term w at position i in document D . If w occurs at position i , it is 1, otherwise 0.
- $k(i, j)$: the propagated count to position i from a term at position j (i.e., w_j). Intuitively, given w_j , $k(i, j)$ serves as a discounting factor and can be any non-increasing function of $|i - j|$, that is, $k(i, j)$ favors positions close to j .



Positional LM

- $c'(w,i)$: the total propagated count of term w at position i from the occurrences of w in all the positions. That is,

$$c'(w,i) = \sum_{j=1}^N d(w,j)k(i,j).$$

- Based on term propagation, we have a term frequency vector $\langle c'(w_1,i), \dots, c'(w_N,i) \rangle$ at position i , forming a virtual document D_i .
- Thus the language model of this virtual document can be estimated as:

$$p(w|D,i) = \frac{c'(w,i)}{\sum_{w \in V} c'(w,i)}$$

where V is the vocabulary set. We call $p(w|D,i)$ a Positional Language Model (PLM) at position i .



Smoothing

- Dirichlet smoothing

$$p_{\mu}(w|D,i) = \frac{c'(w,i) + \mu p(w|C)}{Z_i + \mu}$$

- Jelinek-Mercer smoothing

$$p_{\lambda}(w|D,i) = (1-\lambda)p(w|D,i) + \lambda p(w|C)$$

Kernel functions (term count propagation)

1. Gaussian kernel

$$k(i, j) = \exp\left[-\frac{(i-j)^2}{2\sigma^2}\right]$$

2. Triangle kernel

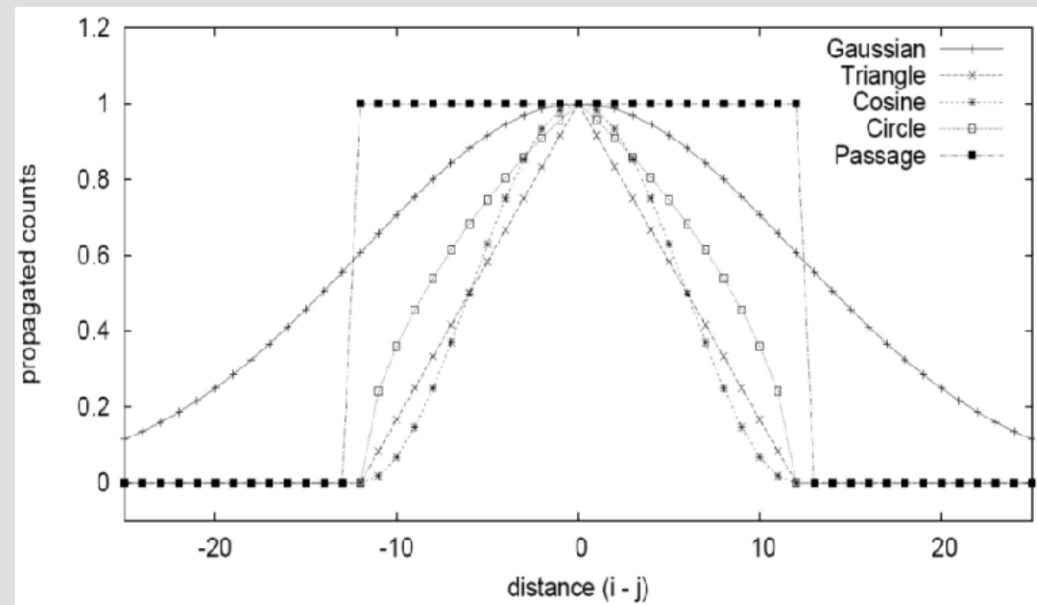
$$k(i, j) = \begin{cases} 1 - \frac{|i-j|}{\sigma} & \text{if } |i-j| \leq \sigma \\ 0 & \text{otherwise} \end{cases}$$

3. Cosine (Hamming) kernel

$$k(i, j) = \begin{cases} \frac{1}{2} \left[1 + \cos\left(\frac{|i-j| \cdot \pi}{\sigma}\right) \right] & \text{if } |i-j| \leq \sigma \\ 0 & \text{otherwise} \end{cases}$$

4. Circle kernel

$$k(i, j) = \begin{cases} \sqrt{1 - \left(\frac{|i-j|}{\sigma}\right)^2} & \text{if } |i-j| \leq \sigma \\ 0 & \text{otherwise} \end{cases}$$



Positional LM – document ranking

1. Best Position Strategy

- The strategy is to simply score a document based on the score of its best matching position:

$$S(Q, D) = \max_{i \in [1, N]} \{S(Q, D, i)\}$$

2. Multi-Position Strategy

- Particularly, we can take the average of the top-k scores to score a document:

$$S(Q, D) = \frac{1}{k} \sum_{i \in TopK} S(Q, D, i)$$

where $TopK$ is the set of positions corresponding to the top-k highest scores of $S(Q, D, i)$.

Positional LM – Document ranking

3. Multi- σ Strategy

- we compute the best position scores for several different values, and then combine these scores together as the final score for a document.

$$\bullet S(Q, D) = \sum_{\sigma \in R} [\beta_{\sigma} \cdot \max \{S_{\sigma}(Q, D, i)\}]$$

where R is a predefined set of σ values, $S_{\sigma}(\cdot)$ is the score function for PLMs with parameter σ , β_{σ} is the weight on different σ ($\sum_{\sigma \in R} \beta_{\sigma} = 1$).

- In particular, if $R = \{\sigma_0, \infty\}$, this strategy equals to an interpolation of the PLM and the regular document language model.



Trick of implementation

- Creating a model for each position is inefficient
- However, $k(i,j) = k(j,i)$, i.e. the count propagated from a position is equal to that to that position
- Equivalent to fixed-length passage retrieval

Experiments with bestposition strategy

- We smooth an estimated PLM when computing retrieval scores. We test both Dirichlet prior smoothing (with parameter 1,000) and Jelinek-Mercer (with parameter 0.5).

| WT2G | | | | | |
|-------------------|---------------|---------------|---------------|---------------|---------------|
| kernel \ σ | 25 | 75 | 125 | 175 | 275 |
| Gaussian | 0.2989 | 0.3213 | 0.3286 | 0.3307 | 0.3285 |
| Triangle | 0.2661 | 0.3028 | 0.3149 | 0.3211 | 0.3288 |
| Cosine | 0.2621 | 0.3007 | 0.3128 | 0.3181 | 0.3243 |
| Circle | 0.2797 | 0.3140 | 0.3225 | 0.3273 | 0.3267 |

| FR | | | | | | |
|----------|----------|---------------|---------------|---------------|---------------|---------------|
| kernel | σ | 25 | 75 | 125 | 175 | 275 |
| Gaussian | | 0.2913 | 0.2679 | 0.2895 | 0.2880 | 0.2846 |
| Triangle | | 0.2585 | 0.2898 | 0.2858 | 0.2682 | 0.2897 |
| Cosine | | 0.2603 | 0.2910 | 0.3000 | 0.2948 | 0.2858 |
| Circle | | 0.2685 | 0.2754 | 0.2673 | 0.2877 | 0.2873 |

| TREC8 | | | | | |
|-------------------|---------------|---------------|---------------|---------------|---------------|
| kernel \ σ | 25 | 75 | 125 | 175 | 275 |
| Gaussian | 0.2364 | 0.2465 | 0.2503 | 0.2535 | 0.2550 |
| Triangle | 0.2244 | 0.2379 | 0.2438 | 0.2475 | 0.2500 |
| Cosine | 0.2257 | 0.2390 | 0.2430 | 0.2457 | 0.2486 |
| Circle | 0.2315 | 0.2401 | 0.2464 | 0.2492 | 0.2523 |

| AP88-89 | | | | | | |
|----------|----------|---------------|---------------|---------------|---------------|---------------|
| kernel | σ | 25 | 75 | 125 | 175 | 275 |
| Gaussian | | 0.1926 | 0.2112 | 0.2162 | 0.2177 | 0.2198 |
| Triangle | | 0.1709 | 0.1987 | 0.2077 | 0.2117 | 0.2173 |
| Cosine | | 0.1682 | 0.1969 | 0.2063 | 0.2107 | 0.2144 |
| Circle | | 0.1801 | 0.2034 | 0.2093 | 0.2135 | 0.2159 |

Dirichlet prior smoothing

| WT2G | | | | | |
|-------------------|---------------|---------------|---------------|---------------|---------------|
| kernel \ σ | 25 | 75 | 125 | 175 | 275 |
| Gaussian | 0.3024 | 0.3170 | 0.3133 | 0.3096 | 0.3010 |
| Triangle | 0.2711 | 0.3057 | 0.3118 | 0.3170 | 0.3131 |
| Cosine | 0.2622 | 0.2855 | 0.2681 | 0.2452 | 0.2039 |
| Circle | 0.2813 | 0.3130 | 0.3188 | 0.3179 | 0.3148 |

| FR | | | | | | |
|----------|----------|---------------|---------------|---------------|---------------|---------------|
| kernel | σ | 25 | 75 | 125 | 175 | 275 |
| Gaussian | | 0.2639 | 0.2606 | 0.2592 | 0.2827 | 0.2822 |
| Triangle | | 0.2458 | 0.2681 | 0.2607 | 0.2610 | 0.2834 |
| Cosine | | 0.2463 | 0.2476 | 0.2424 | 0.2249 | 0.1593 |
| Circle | | 0.2512 | 0.2557 | 0.2613 | 0.2591 | 0.2833 |

| TREC8 | | | | | |
|-------------------|---------------|---------------|---------------|---------------|---------------|
| kernel \ σ | 25 | 75 | 125 | 175 | 275 |
| Gaussian | 0.2454 | 0.2510 | 0.2548 | 0.2575 | 0.2576 |
| Triangle | 0.2335 | 0.2477 | 0.2491 | 0.2506 | 0.2562 |
| Cosine | 0.2335 | 0.2423 | 0.2356 | 0.2227 | 0.2058 |
| Circle | 0.2369 | 0.2456 | 0.2498 | 0.2528 | 0.2555 |

| AP88-89 | | | | | | |
|----------|----------|---------------|---------------|---------------|---------------|---------------|
| kernel | σ | 25 | 75 | 125 | 175 | 275 |
| Gaussian | | 0.1892 | 0.2016 | 0.2054 | 0.2066 | 0.2049 |
| Triangle | | 0.1718 | 0.1933 | 0.1968 | 0.2002 | 0.2051 |
| Cosine | | 0.1701 | 0.1910 | 0.1815 | 0.1636 | 0.1349 |
| Circle | | 0.1735 | 0.1933 | 0.1962 | 0.2010 | 0.2049 |

Jelinek-Mercer smoothing

Experiments – multi σ strategy

- To test this special case of Multi- σ strategy, we fix one value to ∞ , and vary the other one.

| method\data | WT2G | TREC8 | FR | AP88-89 |
|----------------|---------------------------|---------------------------|---------------------|---------------------------|
| KL | 0.2931 | 0.2509 | 0.2697 | 0.2196 |
| $\sigma = 25$ | 0.3247 ⁺ | 0.2562 ⁺ | 0.2936 | 0.2237⁺ |
| $\sigma = 75$ | 0.3336⁺ | 0.2553 ⁺ | 0.2896 ⁺ | 0.2227 |
| $\sigma = 125$ | 0.3330 ⁺ | 0.2559 ⁺ | 0.2885 | 0.2201 |
| $\sigma = 175$ | 0.3324 ⁺ | 0.2574⁺ | 0.2858 | 0.2196 |
| $\sigma = 275$ | 0.3255 ⁺ | 0.2561 ⁺ | 0.2852 | 0.2193 |

- It shows that, when interpolated with document language models, the PLM performs more robustly and effectively.

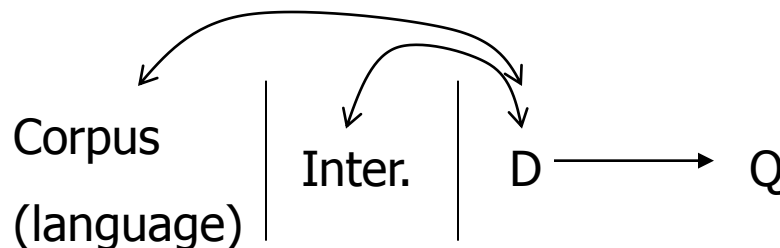
General discussions

- Basic IR models:

- term independence
- Document + corpus (smoothing, $tf \cdot idf$)

- Construct an intermediate model between document and corpus

- Pseudo-relevance feedback: query relevance model
- Document cluster: larger document model



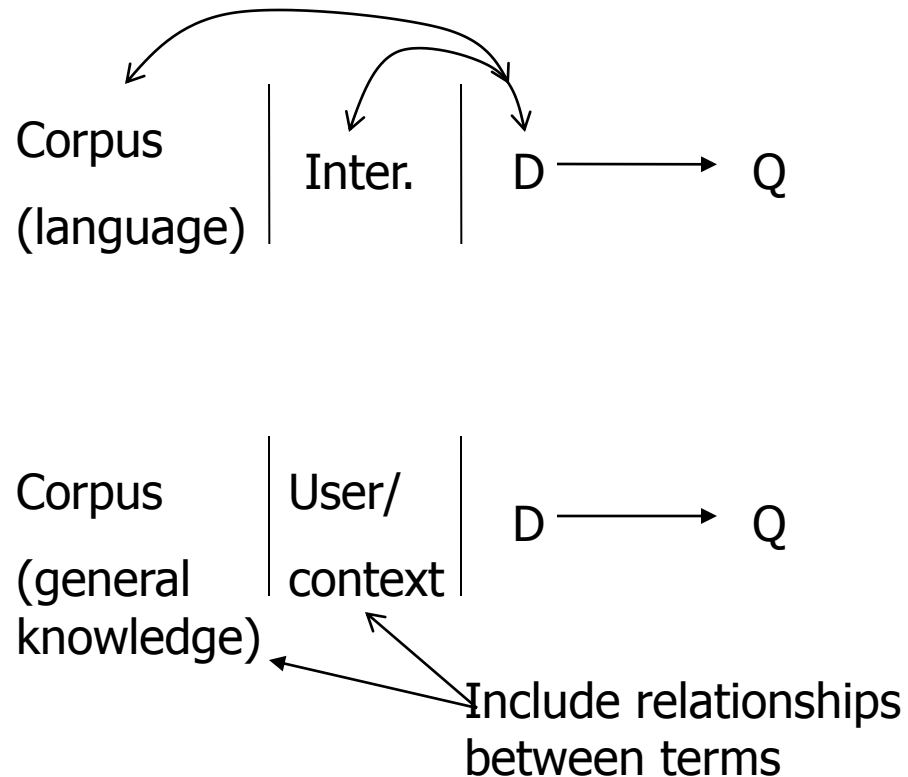


Questions

- The intermediate model only model n-gram distribution
- Add more inference power (domain knowledge)?
 - algorithm → programming
 - $P(\text{programming} \mid \text{algorithm})$
- How?
- Is this helpful?

Challenges

Add user/contextual knowledge





Challenges

- How to estimate
 $P(\text{programming} \mid \text{algorithm})$ or
 $P(\text{algorithm} \rightarrow \text{programming})$?
- How to make it context-dependent?
 - Program \rightarrow computer not suitable in the context of TV, entertainment, ...
- How to integrate?
 - Translation model is insufficient because the translation words are considered independent

$$P(Q \mid D) = \prod_j \sum_{q'_j} P(q_j \mid q'_j) P(q'_j \mid D)$$



Some references

- Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu and Guihong Cao. 2004. Dependence language model for information retrieval. In *SIGIR*.
- Yuanhua Lv and ChengXiang Zhai, *Positional Language Models for Information Retrieval*, in Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 2009.
- Metzler, D. and Croft, W.B., "A Markov Random Field Model for Term Dependencies, » SIGIR 2005, pp. 472-479.
- Lixin Shi, Jian-Yun Nie, Modeling Variable Dependencies between Characters in Chinese Information Retrieval, *AIRS*, 2010, pp. 539-551
- F Song and W B Croft (1999). "[A General Language Model for Information Retrieval](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.6467&rep=rep1&type=pdf)". *Research and Development in Information Retrieval*. pp. 279–280.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.6467&rep=rep1&type=pdf>.
- Munirathnam Srikanth, Rohini Srihari, Biterm Language Models for Document Retrieval, SIGIR, 2002, pp.425-426
- Tao Tao, ChengXiang Zhai, An Exploration of Proximity Measures in Information Retrieval, SIGIR'07, pages 295-302.