

Combining linguistic resources and statistical language modeling for information retrieval

Jian-Yun Nie
RALI, Dept. IRO
University of Montreal, Canada
<http://www.iro.umontreal.ca/~nie>



Brief history of IR and NLP

- Statistical IR (tf*idf)
- Attempts to integrate NLP into IR
 - Identify compound terms
 - Word disambiguation
 - ...
 - Mitigated success
- Statistical NLP
- Trend: integrate statistical NLP into IR (language modeling)

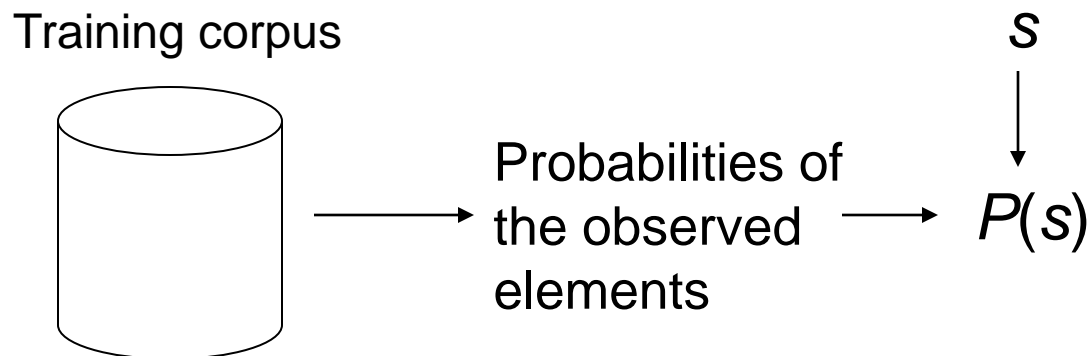


Overview

- Language model
 - Interesting theoretical framework
 - Efficient probability estimation and smoothing methods
 - Good effectiveness
- Limitations
 - Most approaches use uni-grams, and independence assumption
 - Just a different way to weight terms
- Extensions
 - Integrating more linguistic analysis (term relationships)
 - Experiments
- Conclusions

Principle of language modeling

- Goal: create a statistical model so that one can calculate the probability of a sequence of words $S = W_1, W_2, \dots, W_n$ in a language.
- General approach:





Prob. of a sequence of words

$$\begin{aligned} P(s) &= P(w_1, w_2, \dots, w_n) \\ &= P(w_1)P(w_2 | w_1) \dots P(w_n | w_{1,n-1}) \\ &= \prod_{i=1}^n P(w_i | h_i) \end{aligned}$$

Elements to be estimated: $P(w_i | h_i) = \frac{P(h_i w_i)}{P(h_i)}$

- If h_i is too long, one cannot observe (h_i, w_i) in the training corpus, and (h_i, w_i) is hard generalize
- Solution: limit the length of h_i



Estimation

- History: short long
- **modeling:** coarse refined
- **Estimation:** easy difficult
- Maximum likelihood estimation MLE



n-grams

- Limit h_i to $n-1$ preceding words

- Uni-gram:
$$P(s) = \prod_{i=1}^n P(w_i)$$

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

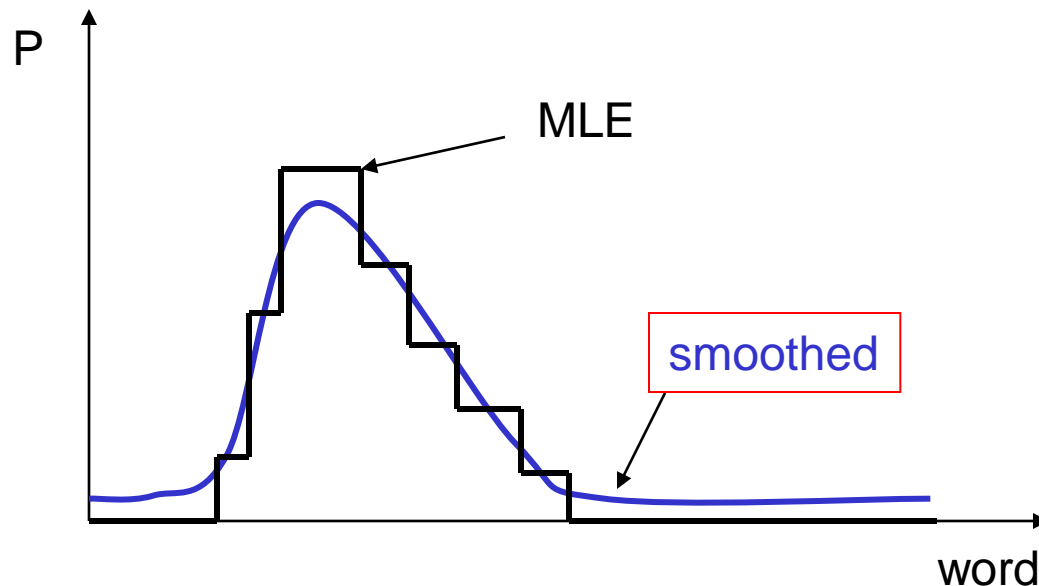
- Tri-gram:
$$P(s) = \prod_{i=1}^n P(w_i | w_{i-2}w_{i-1})$$

- Maximum likelihood estimation MLE

$$P(w_i) = \frac{\#(w_i)}{|C_{uni}|} \quad P(h_i w_i) = \frac{\#(h_i w_i)}{|C_{n-gram}|} \quad \text{problem: } P(h_i w_i) = 0$$

Smoothing

- Goal: assign a low probability to words or n-grams not observed in the training corpus





Smoothing methods

n-gram: α

- Change the freq. of occurrences
 - Laplace smoothing (add-one):

$$P_{add_one}(\alpha | C) = \frac{|\alpha| + 1}{\sum_{\alpha_i \in V} (|\alpha_i| + 1)}$$

- Good-Turing

change the freq. r to $r^* = (r + 1) \frac{n_{r+1}}{n_r}$

n_r = no. of n-grams of freq. r



Smoothing (cont'd)

- Combine a model with a lower-order model

- Backoff (Katz)

$$P_{Katz}(w_i | w_{i-1}) = \begin{cases} P_{GT}(w_i | w_{i-1}) & \text{if } |w_{i-1}w_i| > 0 \\ \alpha(w_{i-1})P_{Katz}(w_i) & \text{otherwise} \end{cases}$$

- Interpolation (Jelinek-Mercer)

$$P_{JM}(w_i | w_{i-1}) = \lambda_{w_{i-1}} P_{ML}(w_i | w_{i-1}) + (1 - \lambda_{w_{i-1}}) P_{JM}(w_i)$$

- In IR, combine doc. with corpus

$$P(w_i | D) = \lambda P_{ML}(w_i | D) + (1 - \lambda) P_{ML}(w_i | C)$$



Smoothing (cont'd)

- Dirichlet

$$P_{Dir}(w_i | D) = \frac{tf(w_i, D) + \mu P_{ML}(w_i | C)}{|D| + \mu}$$

- Two-stage

$$P_{TS}(w_i | D) = (1 - \lambda) \frac{tf(w_i, D) + \mu P_{ML}(w_i | C)}{|D| + \mu} + \lambda P_{ML}(w_i | C)$$



Using LM in IR

- Principle 1:
 - Document D: Language model $P(w|M_D)$
 - Query Q = sequence of words q_1, q_2, \dots, q_n (uni-grams)
 - Matching: $P(Q|M_D)$
- Principle 2:
 - Document D: Language model $P(w|M_D)$
 - Query Q: Language model $P(w|M_Q)$
 - Matching: comparison between $P(w|M_D)$ and $P(w|M_Q)$
- Principle 3:
 - Translate D to Q



Principle 1: Document LM

- Document D: Model M_D
- Query Q: q_1, q_2, \dots, q_n : uni-grams
- $P(Q|D) = P(Q|M_D)$
 $= P(q_1|M_D) P(q_2|M_D) \dots P(q_n|M_D)$
- Problem of smoothing
 - Short document
 - Coarse M_D
 - Unseen words

Smoothing

- Change word freq.
- Smooth with corpus

Exemple $P(w_i | D) = \lambda P_{GT}(w_i | D) + (1 - \lambda) P_{ML}(w_i | C)$



Determine λ_i

$$P(w_i) = \lambda_1 P_1(w_i) + \lambda_2 P_2(w_i) \text{ with } \lambda_1 + \lambda_2 = 1$$

- Expectation maximization (EM): Choose λ_i that maximizes the likelihood of the text

- Initialize λ_i

- E-step

$$C_i = \sum_w \frac{\lambda_i P_i(w)}{\sum_j \lambda_j P_j(w)}$$

- M-step

$$\lambda_i = \frac{C_i}{\sum_j C_j}$$

- Loop on E and M



Principle 2: Doc. likelihood / divergence between M_d and M_Q

Question: Is the document likelihood increased when a query is submitted?

$$LR(D, Q) = \frac{P(D | Q)}{P(D)} = \frac{P(Q | D)}{P(Q)}$$

(Is the query likelihood increased when D is retrieved?)

- $P(Q|D)$ calculated with $P(Q|M_D)$
- $P(Q)$ estimated as $P(Q|M_C)$

$$Score(Q, D) = \log \frac{P(Q | M_D)}{P(Q | M_C)}$$

Divergence of M_D and M_Q

Assume Q follows a multinomial distribution :

$$P(Q | M_D) = \frac{|Q|!}{\prod_{q_i \in Q} tf(q_i, Q)!} \prod_{q_i \in Q} P(q_i | D)^{tf(q_i, Q)}$$

$$P(Q | M_C) = \frac{|Q|!}{\prod_{q_i \in Q} tf(q_i, Q)!} \prod_{q_i \in Q} P(q_i | C)^{tf(q_i, Q)}$$

$$\begin{aligned} \text{Score}(Q, D) &= \sum_{i=1}^n tf(q_i, Q) * \log \frac{P(q_i | M_D)}{P(q_i | M_C)} \\ &\propto \sum_{i=1}^n P(q_i | M_Q) * \log \frac{P(q_i | M_D)}{P(q_i | M_C)} \\ &= \sum_{i=1}^n P(q_i | M_Q) * \log \frac{P(q_i | M_D)}{P(q_i | M_Q)} - \sum_{i=1}^n P(q_i | M_Q) * \log \frac{P(q_i | M_C)}{P(q_i | M_Q)} \\ &= -KL(M_Q, M_D) + \text{Constant} = H(M_Q | M_C) - H(M_Q | M_D) \end{aligned}$$

KL: Kullback-Leibler divergence, measuring the divergence of two probability distributions

Principle 3: IR as translation

Noisy channel: *message*  *received*

- Transmit D through the channel, and receive Q

$$P(Q | D) = \prod_i P(q_i | D) = \prod_i \sum_j P(q_i | w_j) P(w_j | D)$$

- $P(w_j | D)$: prob. that D generates w_j
- $P(q_i | w_j)$: prob. of translating w_j by q_i
- Possibility to consider relationships between words
- How to estimate $P(q_i | w_j)$?
 - Berger&Lafferty: Pseudo-parallel texts (align sentence with paragraph)



Summary on LM

- Can a query be generated from a document model?
- Does a document become more likely when a query is submitted (or reverse)?
- Is a query a "translation" of a document?

- Smoothing is crucial
- Often use uni-grams



Beyond uni-grams

- Bi-grams

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

- Bi-term

- Do not consider word order in bi-grams

(analysis, data) – (data, analysis)



Relevance model

- LM does not capture “Relevance”
- Using pseudo-relevance feedback
 - Construct a “relevance” model using top-ranked documents
- Document model + relevance model (feedback) + corpus model



Experimental results

- LM vs. Vector space model with $tf*idf$ (Smart)
 - Usually better
- LM vs. Prob. model (Okapi)
 - Often similar
- bi-gram LM vs. uni-gram LM
 - Slight improvements (but with much larger model)



Contributions of LM to IR

- Well founded theoretical framework
- Exploit the mass of data available
- Techniques of smoothing for probability estimation
- Explain some empirical and heuristic methods by smoothing
- Interesting experimental results
- Existing tools for IR using LM (Lemur)



Problems

- Limitation to uni-grams:
 - No dependence between words
- Problems with bi-grams
 - Consider all the adjacent word pairs (noise)
 - Cannot consider more distant dependencies
 - Word order – not always important for IR
- Entirely data-driven, no external knowledge
 - e.g. programming → computer
- Logic well hidden behind numbers
 - Key = smoothing
 - Maybe too much emphasis on smoothing, and too little on the underlying logic
- Direct comparison between D and Q
 - Requires that D and Q contain identical words (except translation model)
 - Cannot deal with synonymy and polysemy



Some Extensions

- Classical LM:

Document \rightarrow t_1, t_2, \dots \rightarrow Query
(ind. terms)

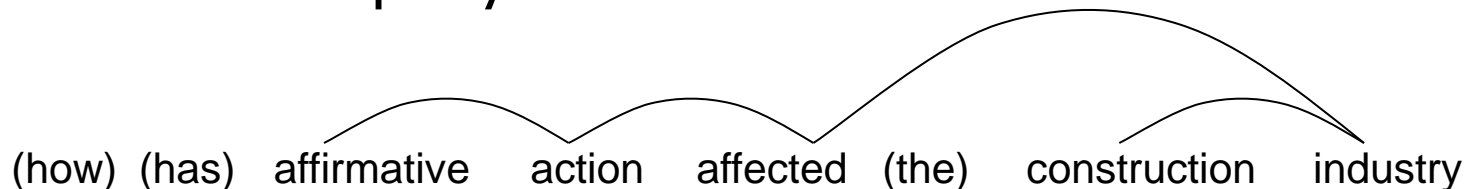
1. Document \rightarrow comp.archi. \rightarrow Query
(dep. terms)

2. Document \rightarrow prog. \rightarrow comp. \rightarrow Query
(term relations)

Extensions (1): link terms in document and query

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

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



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

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)

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

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

$$P(Q | L, D) = P(q_h | D) \prod_{(i,j) \in L} P(q_j | q_i, L, D) = \dots$$

$$= \prod_{i=1..n} P(q_i | D) \prod_{(i,j) \in L} \frac{P(q_i, q_j | L, D)}{P(q_i | D)P(q_j | D)}$$

Requirement on **words** and **bi-terms**

Experiments

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).



Extension (2): Inference in IR

- Logical deduction

$$(A \rightarrow B) \wedge (B \rightarrow C) \mid - A \rightarrow C$$

- In IR: D=Tsunami, Q=natural disaster

$$(D \rightarrow Q') \wedge (Q' \rightarrow Q) \mid - D \rightarrow Q$$

$\underbrace{(D \rightarrow Q')}$ $\underbrace{(Q' \rightarrow Q)}$
Direct matching **Inference** on query

$$(D \rightarrow D') \wedge (D' \rightarrow Q) \mid - D \rightarrow Q$$

$\underbrace{(D \rightarrow D')}$ $\underbrace{(D' \rightarrow Q)}$
Inference on doc. Direct matching



Is LM capable of inference?

- Generative model: $P(Q|D)$
- $P(Q|D) \sim P(D \rightarrow Q)$
- Smoothing:

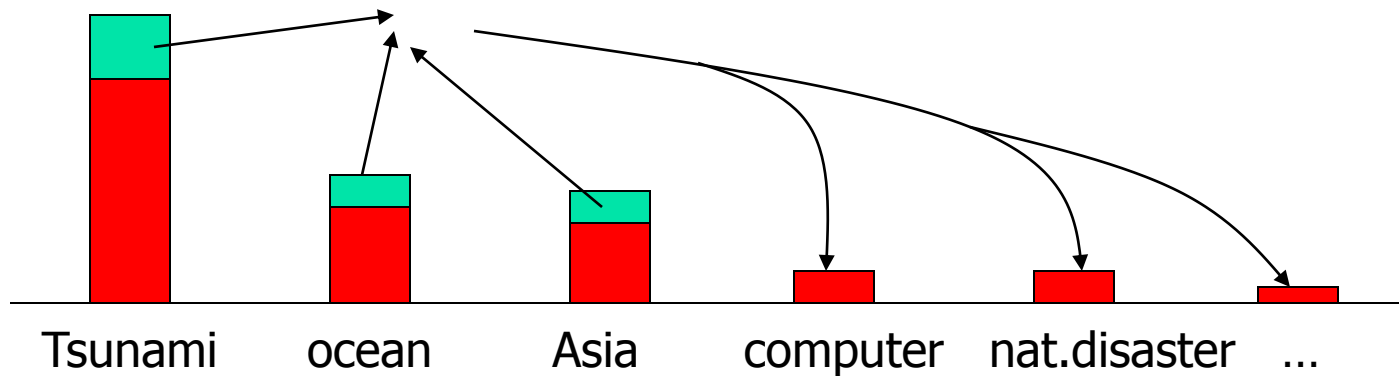
$$P(t_i | D) = \lambda P_{ML}(t_i | D) + (1 - \lambda) P_{ML}(t_i | C)$$

$$t_i \notin D : P_{ML}(t_i | D) = 0$$

change to $P(t_i | D) > 0$

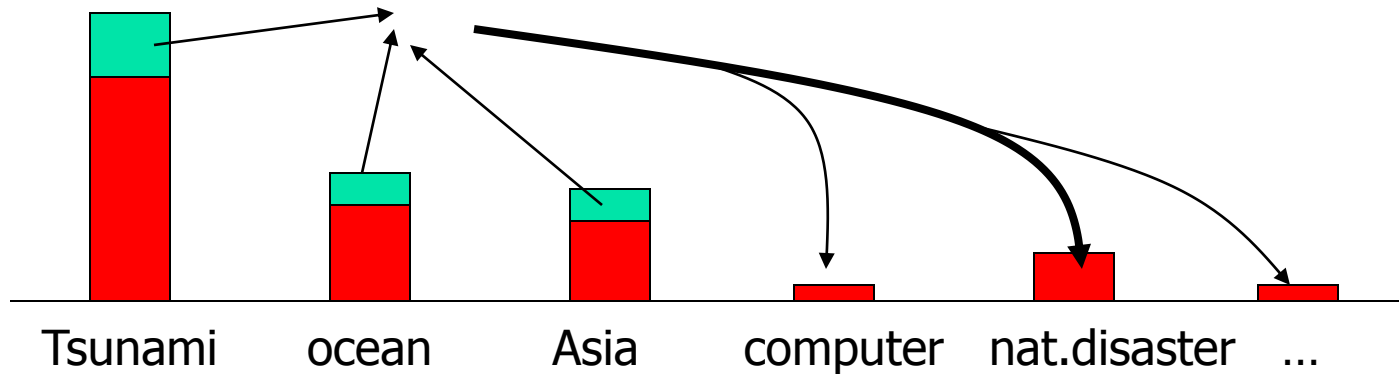
- E.g. $D = \text{Tsunami}$, $P_{ML}(\text{natural disaster} | D) = 0$
change to $P(\text{natural disaster} | D) > 0$
- No inference
 - $P(\text{computer} | D) > 0$

Effect of smoothing?



- Smoothing \neq inference
- Redistribution uniformly/according to collection

Expected effect

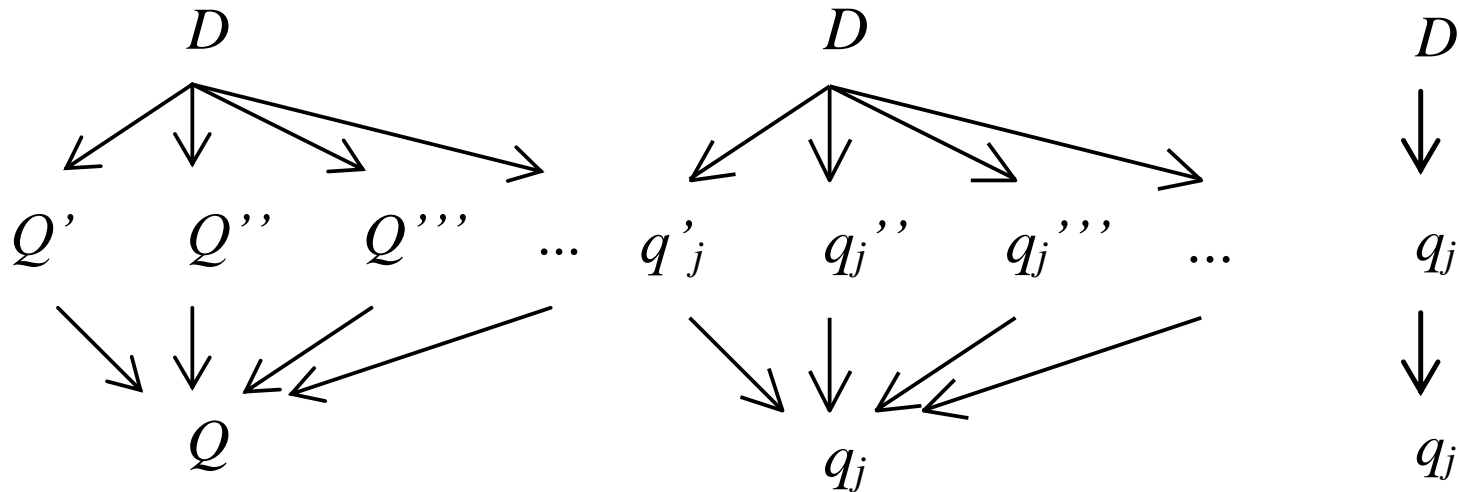


- Using Tsunami → natural disaster
- Knowledge-based smoothing

Extended translation model

$$(D \rightarrow Q') \wedge (Q' \rightarrow Q) \mid -(D \rightarrow Q)$$

$$(D \rightarrow t_j) \wedge (t_j \rightarrow t_i) \mid -(D \rightarrow t_i)$$



Translation model:
$$P(q_j | D) = \sum_{q'_j} P(q_j | q'_j) P(q'_j | D)$$

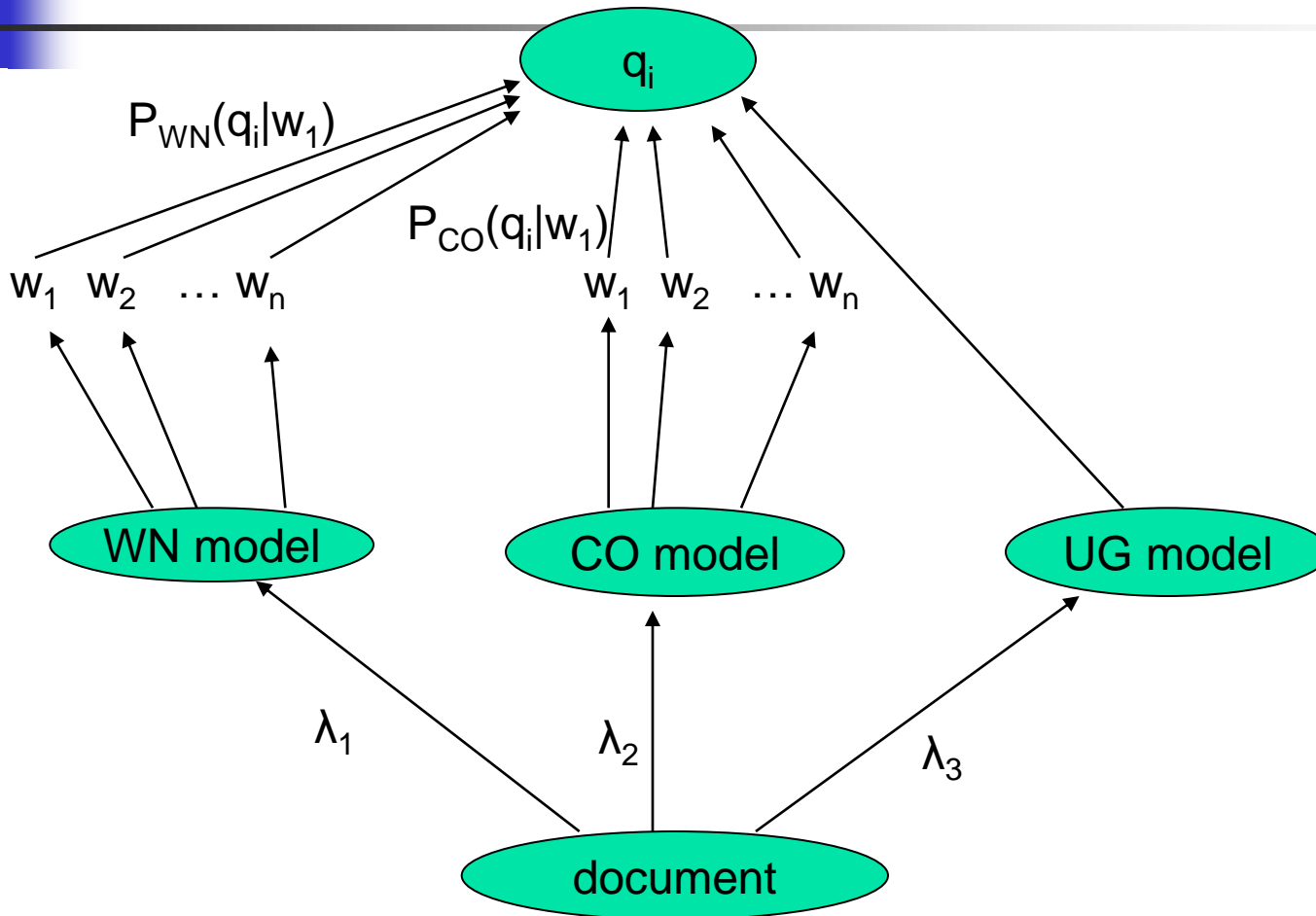
$$P(Q | D) = \prod_j \sum_{q'_j} P(q_j | q'_j) P(q'_j | D)$$

Using other types of knowledge?

- Different ways to satisfy a query (q. term)
 - Directly through unigram model
 - Indirectly (by inference) through Wordnet relations
 - Indirectly through Co-occurrence relations
 - ...
- $D \rightarrow t_i$ if $D \rightarrow_{UG} t_i$ or $D \rightarrow_{WN} t_i$ or $D \rightarrow_{CO} t_i$

$$P(t_i | D) = \lambda_1 \sum_j P_{WN}(t_i | t_j) P(t_j | D) + \lambda_2 \sum_j P_{CO}(t_i | t_j) P(t_j | D) + \lambda_3 P_{UG}(t_i | C)$$

Illustration (Cao et al. 05)





Experiments

Table 3: Different combinations of unigram model, link model and co-occurrence model

Model	WSJ		AP		SJM	
	AvgP	Rec.	AvgP	Rec.	AvgP	Rec.
UM	0.2466	1659/2172	0.1925	3289/6101	0.2045	1417/2322
CM	0.2205	1700/2172	0.2033	3530/6101	0.1863	1515/2322
LM	<i>0.2202</i>	<i>1502/2172</i>	<i>0.1795</i>	<i>3275/6101</i>	<i>0.1661</i>	<i>1309/2322</i>
UM+CM	0.2527	1700/2172	0.2085	3533/6101	0.2111	1521/2322
UM+LM	0.2542	1690/2172	0.1939	3342/6101	0.2103	1558/2332
UM+CM+LM	<i>0.2597</i>	<i>1706/2172</i>	<i>0.2128</i>	<i>3523/6101</i>	<i>0.2142</i>	<i>1572/2322</i>

UM=Unigram, CM=co-occ. model, LM=model with Wordnet

Experimental results

Coll.	Unigram Model		Dependency Model					
			LM with unique WN rel.			LM with typed WN rel.		
	AvgP	Rec.	AvgP	%change	Rec.	AvgP	%change	Rec.
WSJ	0.2466	1659/2172	0.2597	+5.31*	1706/2172	0.2623	+6.37*	1719/2172
AP	0.1925	3289/6101	0.2128	+10.54**	3523/6101	0.2141	+11.22**	3530/6101
SJM	0.2045	1417/2322	0.2142	+4.74	1572/2322	0.2155	+5.38	1558/2322

Integrating different types of relationships in LM may improve effectiveness

Doc expansion v.s. Query expansion

$$P(t_i | Q) = P_{UG}(t_i | D)$$

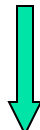


Document expansion

$$P(t_i | D) = \sum_{t_j} P(t_i | t_j) P(t_j | D)$$

$$P(t_i | D) = \lambda_1 \sum_{t_j} P_{WN}(t_i | t_j) P(t_j | D) + \lambda_2 \sum_{t_j} P_{CO}(t_i | t_j) P(t_j | D) + \lambda_3 P_{UG}(t_i | D)$$

$$P(t_i | D) = P_{UG}(t_i | Q)$$



Query expansion

$$P(t_i | Q) = \sum_{t_j} P(t_i | t_j) P(t_j | Q)$$

$$P(t_i | Q) = \lambda_1 \sum_{t_j} P_R(t_i | t_j) P(t_j | Q) + \lambda_2 P_{UG}(t_j | Q)$$



Implementing QE in LM

- KL divergence:

$$\begin{aligned} \text{Score}(Q, D) &= -KL(Q; D) = \sum_{t_i \in Q} P(t_i | Q) \log \frac{P(t_i | D)}{P(t_i | Q)} \\ &= \sum_{t_i \in Q} P(t_i | Q) \log P(t_i | D) - \sum_{t_i \in Q} P(t_i | Q) \log P(t_i | Q) \\ &\propto \sum_{t_i \in Q} P(t_i | Q) \log P(t_i | D) \end{aligned}$$

Query expansion = a new $P(t_i | Q)$



Expanding query model

$$P(q_i | Q) = \lambda P_{ML}(q_i | Q) + (1 - \lambda) P_R(q_i | Q)$$

$P_{ML}(t_j | Q)$: Max.Likelihood unigram model (not smoothed)

$P_R(t_i | Q)$: Relational model

$$Score(Q, D) = \sum_{q_i \in V} P(q_i | Q) \times \log P(q_i | D)$$

$$= \sum_{q_i \in V} [\lambda P_{ML}(q_i | Q) + (1 - \lambda) P_R(q_i | Q)] \times \log P(q_i | D)$$

$$= \lambda \sum_{q_i \in Q} P_{ML}(q_i | Q) \times \log P(q_i | D) + (1 - \lambda) \sum_{q_i \in V} P_R(q_i | Q) \times \log P(q_i | D)$$

Classical LM

Relation model



How to estimate $P_R(t_i | Q)$?

- Using co-occurrence information
- Using an external knowledge base (e.g. Wordnet)
- Pseudo-rel. feedback
- Other term relationships
- ...



Defining relational model

- HAL (Hyperspace Analogue to Language): a special co-occurrence matrix (Bruza&Song)
 - “the effects of pollution on the population”
-
-
-
-

“effects” and “pollution” co-occur in 2 windows ($L=3$)

$$\text{HAL}(\text{effects}, \text{pollution}) = 2 = L - \text{distance} + 1$$



From HAL to Inference relation

$$P_{HAL}(t_2 | t_1) = \frac{HAL(t_1, t_2)}{\sum_{t_i} HAL(t_1, t_i)}$$

- ***superconductors*** : <U.S.:0.11, american:0.07, basic:0.11, bulk:0.13 ,called:0.15, capacity:0.08, carry:0.15, ceramic:0.11, commercial:0.15, consortium:0.18, cooled:0.06, current:0.10, develop:0.12, dover:0.06, ...>
- Combining terms: *space*⊕*program*
 - Different importance for *space* and *program*

From HAL to Inference relation (information flow)

$$\text{degree}(t_{i_1}, \dots, t_{i_n} | - t_j) = \text{degree}(\oplus t_i | - t_j) = \frac{P(\oplus t_i, t_j)}{\sum_{t_k \in QP(\oplus t_i)} P(\oplus t_i, t_k)}$$

$$P_{IF}(t_{i_1}, \dots, t_{i_n} | - t_j) = \frac{\text{degree}(t_{i_1}, \dots, t_{i_n} | - t_j)}{\sum_{t_k \in V} \text{degree}(t_{i_1}, \dots, t_{i_n} | - t_k)}$$

- **space** \oplus **program** | - {program:1.00 space:1.00 nasa:0.97 new:0.97 U.S.:0.96 agency:0.95 shuttle:0.95 ... science:0.88 **scheduled:0.87 reagan:0.87** director:0.87 programs:0.87 air:0.87 **put:0.87** center:0.87 billion:0.87 aeronautics:0.87 **satellite:0.87, ...>**



Two types of term relationship

- Pairwise $P(t_2|t_1)$:
$$P_{HAL}(t_2 | t_1) = \frac{HAL(t_1, t_2)}{\sum_{t_i} HAL(t_1, t_i)}$$
- Inference relationship

$$P_{IF}(t_{i_1}, \dots, t_{i_n} | -t_j) = \frac{\text{degree}(t_{i_1}, \dots, t_{i_n} | -t_j)}{\sum_{t_k \in V} \text{degree}(t_{i_1}, \dots, t_{i_n} | -t_k)}$$

- Inference relationships are less ambiguous and produce less noise (Qiu&Frei 93)

1. Query expansion with pairwise term relationships

$$\begin{aligned} \text{Score}(Q, D) &= \lambda \sum_{q_i \in Q} P_{ML}(q_i | Q) \times \log P(q_i | D) + (1 - \lambda) \sum_{q_i \in V} P_R(q_i | Q) \times \log P(q_i | D) \\ &= \lambda \sum_{q_i \in Q} P_{ML}(q_i | Q) \times \log P(q_i | D) \\ &\quad + (1 - \lambda) \sum_{q_i \in V} \sum_{q_j \in Q} P_{co}(q_i | q_j) \times P(q_j | Q) \times \log P(q_i | D) \\ &\approx \lambda \sum_{q_i \in Q} P_{ML}(q_i | Q) \times \log P(q_i | D) \\ &\quad + (1 - \lambda) \sum_{q_j \in Q \wedge R(q_i, q_j) \in E} P_{co}(q_i | q_j) \times P(q_j | Q) \times \log P(q_i | D) \end{aligned}$$

Select a set (85) of strongest HAL relationships

2. Query expansion with IF term relationships

$$\begin{aligned} \text{Score}(Q, D) &= \lambda \sum_{q_i \in Q} P_{ML}(q_i | Q) \times \log P(q_i | D) + (1 - \lambda) \sum_{q_i \in V} P_R(q_i | Q) \times \log P(q_i | D) \\ &= \lambda \sum_{q_i \in Q} P_{ML}(q_i | Q) \times \log P(q_i | D) \\ &\quad + (1 - \lambda) \sum_{q_i \in V} \sum_{Q_j \in Q} P_{IF}(q_i | Q_j) \times P(Q_j | Q) \times \log P(q_i | D) \\ &\approx \lambda \sum_{q_i \in Q} P_{ML}(q_i | Q) \times \log P(q_i | D) \\ &\quad + (1 - \lambda) \sum_{Q_j \in Q \wedge R(q_i, Q_j) \in E} P_{IF}(q_i | Q_j) \times P(Q_j | Q) \times \log P(q_i | D) \end{aligned}$$

85 strongest IF relationships

Experiments (Bai et al. 05)

(AP89 collection, query 1-50)

	Doc. Smooth.	LM baseline	QE with HAL	QE with IF	QE with IF & FB
AvgPr	Jelinek-Merer	0.1946	0.2037 (+5%)	0.2526 (+30%)	0.2620 (+35%)
	Dirichlet	0.2014	0.2089 (+4%)	0.2524 (+25%)	0.2663 (+32%)
	Absolute	0.1939	0.2039 (+5%)	0.2444 (+26%)	0.2617 (+35%)
	Two-Stage	0.2035	0.2104 (+3%)	0.2543 (+25%)	0.2665 (+31%)
Recall	Jelinek-Merer	1542/3301	1588/3301 (+3%)	2240/3301 (+45%)	2366/3301 (+53%)
	Dirichlet	1569/3301	1608/3301 (+2%)	2246/3301 (+43%)	2356/3301 (+50%)
	Absolute	1560/3301	1607/3301 (+3%)	2151/3301 (+38%)	2289/3301 (+47%)
	Two-Stage	1573/3301	1596/3301 (+1%)	2221/3301 (+41%)	2356/3301 (+50%)

Experiments

(AP88-90, topics 101-150)

	Doc. Smooth.	LM baseline	QE with HAL	QE with IF	QE with IF & FB
AvgPr	Jelinek-Mercer	0.2120	0.2235 (+5%)	0.2742 (+29%)	0.3199 (+51%)
	Dirichlet	0.2346	0.2437 (+4%)	0.2745 (+17%)	0.3157 (+35%)
	Abslute	0.2205	0.2320 (+5%)	0.2697 (+22%)	0.3161 (+43%)
	Two-Stage	0.2362	0.2457 (+4%)	0.2811 (+19%)	0.3186 (+35%)
Recall	Jelinek-Mercer	3061/4805	3142/3301 (+3%)	3675/4805 (+20%)	3895/4805 (+27%)
	Dirichlet	3156/4805	3246/3301 (+3%)	3738/4805 (+18%)	3930/4805 (+25%)
	Abslute	3031/4805	3125/3301 (+3%)	3572/4805 (+18%)	3842/4805 (+27%)
	Two-Stage	3134/4805	3212/3301 (+2%)	3713/4805 (+18%)	3901/4805 (+24%)



Observations

- Possible to implement query/document expansion in LM
- Expansion using inference relationships is more context-sensitive: Better than context-independent expansion (Qiu&Frei)
- Every kind of knowledge always useful (co-occ., Wordnet, IF relationships, etc.)
- LM with some inferential power



Conclusions

- LM = suitable model for IR
- Classical LM = independent terms (n-grams)
- Possibility to integrate linguistic resources:
 - Term relationships:
 - Within document and within query (link constraint \sim compound term)
 - Between document and query (inference)
 - Both
- Automatic parameter estimation = powerful tool for data-driven IR
- Experiments showed encouraging results
- IR works well with statistical NLP
- More linguistic analysis for IR?