STS2009
2nd Search Technology Summit

Proximity Language Model

A Language Model beyond Bag of Words through Proximity

Jinglei Zhao[1] Yeogirl Yun[1,2]

[1] iZENEsoft, Inc.

[2] Wisenut, Inc.

STS2009
2nd Search Technology Summit

Outline

Ⅰ Introduction

Ⅱ The proposed model

- Proximity Language Model
- Modeling Proximate Centrality of Terms

Ⅲ Experiment and Result

Introduction

Background

Probabilistic models are prevalent in IR.

- Documents are represented as “bag of words” (BOW).
- Statistics usually exploited under BOW:
  - Term frequency,inverse document frequency
  - Document length, etc.
- Merits
  - Simplicity in modeling.
  - Effectiveness in parameter estimation.

Model more under the BOW assumption.

- BOW are criticized for not capturing the relatedness between terms.
- Could we model term relatedness while retain the simplicity of probabilistic modeling under BOW?

Introduction

Background

Proximity information.

- Represents the closeness or compactness of the query terms appearing in a document.
- Underlying intuition of using proximity in ranking:
  - The more compact the terms, the more likely that they are topically related.
  - The closer the query terms appear, the more possible the document is relevant.
- It can be seen as a kind of indirect measure of term relatedness or dependence.

Introduction

Objective

Integrate proximity information into Unigram language modeling.

- Language modeling has become a very promising direction in IR.
  - Solid theoretical background.
  - Empirical good performance.
- This paper's focus:
  - Develop a systematic way to integrate the term proximity information into the unigram language modeling.

Introduction

Related Work

Dependency Modeling

- General language model, dependency language model,etc.
- Shortcoming: The parameter estimation become much more difficult to compute and sensitive to data sparse and noise.

Phrase Indexing

- Incorporate bigger unit than word such as phrase or loose phrase in text representation.
- Shortcoming: The improvement of using phrases is not consistent.

Previous Proximity Modeling

- Span-based, pair-based.
- Shortcoming: Combining with relevant score at document-level, intuitive, without theoretical ground.

Introduction

Our Approach

Integrate Proximity with Unigram Language Model

- View query term's proximate centrality as Dirichlet hyper-parameters.
- Combines the score at the term level.
- Boost a term's score contribution when the term is at a central place in the proximity structure.

Merits

- A uniform ranking formula.
- Mathematically grounded.
- Performs better empirically.

Proximity Language Model

Unigram Language Model

Represent query and document as vectors of term counts

$$q = (q_1, q_2, \ldots, q_{|V|})$$
$$d_l = (d_{l,1}, d_{l,2}, \ldots, d_{l,|V|})$$

Query and document are generated by multinomial distribution

$$p(q|\theta_l) = \frac{n_q!}{\prod_{i=1}^{|V|} q_i!} \prod_{i=1}^{|V|} (\theta_{l,i})^{q_i}$$

The relevance of d_l to q is measured by the probability of generating q by the language model estimated from d_l

$$\hat{\theta}_{l,i}^{MLE} = \frac{d_{l,i}}{n_l}$$

Proximity Language Model

Integration with Proximity

Our belief and expectation

- *Given d_a and d_b, supposing all others being equal while the query terms of q appears more proximate in d_a than in d_b, we believe that d_a should be more relevant to the query than d_b.*
- *In other words, if $\hat{\theta}_a$ and $\hat{\theta}_b$ represent the language model estimated from d_a and d_b respectively, we believe that the probability that q is generated from $\hat{\theta}_a$ should be higher than $\hat{\theta}_b$.*

Express our expectation

- *Term's emission probability $\theta_{l,i}$ should be proportional to each term's proximity centraility score $Prox_{d_l}(w_i)$ with respect to other query terms.*
- *View $Prox_{d_1}(w_i)$ as the weight on $\theta_{l,i}$.*
- *Express the above two points by using a conjugate prior on θ_l.*

Proximity Language Model

Integration with Proximity

Dirichlet prior on θ_l

$$p(\theta_l|u) = Z_u \prod_{i=1}^{|V|} (\theta_{l,i})^{u_i - 1} \equiv Dir(\theta_l|u)$$

where $u_i = \lambda Prox(w_i)$.

The posterior estimation of θ_l

$$p(\theta_l|d_l, u) = \frac{p(d_l|\theta_l)p(\theta_l|u)}{p(d_l|u)} = Z_{u'} \prod_{i=1}^{|V|} (\theta_{l,i})^{d_{l,i}+u_i-1}$$

The proximity integrated estimation of the word emission probability

$$\theta_{l,i}^{P} = \int_{\theta_l} Dir(\theta_l|u + d_l)\theta_l d\theta_l = \frac{d_{l,i} + \lambda Prox_{d_l}(w_i)}{n_l + \sum_{i=1}^{|V|} \lambda Prox_{d_l}(w_i)}$$

Proximity Language Model

Integration with Proximity

Interpretation on proximity document model

- Transform proximity information to word count information.
- Boost a term's likelihood when it is proximate to other terms.
- From the original bag of words to a pseudo "bag of words".
- More generally, a way of model term relatedness under BOW?

Relation with smoothing.

- The proximity factor mainly functions to adjust the parameters for seen
- matching terms with respect to a query in a document.
- Smoothing is motivated to weight the unseen words in the document.

Proximity Language Model

Integration with Proximity

Further smoothing with collection language model

$$\hat{\theta}^{P}_{l,i} = \frac{d_{l,i} + \lambda Prox_{d_l}(w_i) + \mu p(w_i|C)}{n_l + \sum_{i=1}^{|V|} \lambda Prox_{d_l}(w_i) + \mu}$$

The ranking formula under KL divergence framework

$$Rank(q, d_l) = \sum_{i:d_{l,i}>0, q_i>0} p(w_i|\hat{\theta}_q) log \frac{p_s(w_i|d_l, u)}{\alpha_d p(w_i|C)} + log \alpha_d$$

where, $p_s(w_i|d_l, u) = \hat{\theta}^{P}_{l,i}$, $\alpha_d = \frac{\mu}{n_l + \sum_{i=1}^{|V|} \lambda Prox_{d_l}(w_i) + \mu}$

Term Proximity Measure

Term's Proximate centrality

- *A key notion in PLM is the estimation of term proximity : $Prox_{d_l}(w_i)$.*
- *For non - query terms, they are assumed to have a constant score of zero.*
- *For a query term, it should be computed according to a proximity measure that reflects the terms closeness to other query term' s.*

Measuring Proximity via Pair Distance

- Represent a term's proximity by measuring its distance to other query terms in the document.
- How to define a term's distance to other terms in a document?
- how to map term distance to the term's proximate centrality score?

Term Proximity Measure

Pairwise term distance

Represented as the distance between the closest occurring positions of the two terms in the document.

$$Dis(Q_i, Q_j; D) = \begin{cases} |D|, \text{ if } |O_{Q_i}| = 0 \text{ or } |O_{Q_j}| = 0 \\ MIN\_DIS, \text{ otherwise} \end{cases}$$

$$MIN\_DIS = min_{o_{i_k} \in O_{Q_i}, o_{j_k} \in O_{Q_j}} \{Dis(o_{i_k}, o_{j_k}; D)\}$$

Pairwise proximity

$$Prox(Q_i, Q_j; D) = f(Dis(Q_i, Q_j; D))$$

$$f = para^{-data}$$

Computation of Term's Proximate Centrality

Term Proximity based on Minimum Distance

$$P\_MinDist(Q_i) = f(min_{Q_j \neq Q_i, Q_j \in Q}\{Dis(Q_i, Q_j; D)\})$$

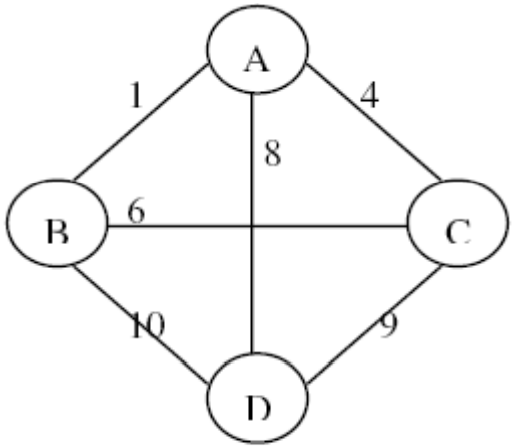
Term Proximity based on Average Distance

$$P\_AveDist(Q_i) = f(\frac{1}{n-1}\sum_{Q_j \neq Q_i, Q_j \in Q} Dis(Q_i, Q_j; D))$$

Term Proximity Summed over Pair Proximity

$$P\_SumProx(Q_i) = \sum_{Q_j \neq Q_i, Q_j \in Q} f(Dis(Q_i, Q_j; D))$$

An example



Proximity computed by different measures ($f = 1.5^{-dist}$)

Measure	Prox(A)	Prox(B)	Prox(C)	Prox(D)
P_MinDist	0.67	0.67	0.20	0.04
P_AveDist	0.17	0.10	0.06	0.03
P_SumProx	0.91	0.78	0.26	0.08

Experiment and Result

Experimental Setting

Data Set

Collection	#Doc.	Len.	Queries	#qrel.
AP88	79919	488	TOPIC251-300	1672
WSJ90-92	74520	514	TOPIC251-300	1064
WSJ87-92	173252	473	TOPIC151-200	3913
OHSUMED	348566	129	Ohsumed topic	3875

Experimental platform

- Lemur toolkit.
- A naive tokenizer.
- A very small stopword list.

Experiment and Result

Experimental Setting

Baselines

- Basic KL divergence language model (LM)

$$Rank_{LM}(q,d) = \sum_{i:q_i>0,d_i>0} q_i \cdot log(1 + \frac{d_i}{\mu \cdot p(w|C)}) + |q| \cdot log \frac{\mu}{|d| + \mu}$$

- Tao's document-level linear score combination (LLM).

$$Rank(Q,D) = Rank_{LM}(q,d) + log(\gamma + exp(-\delta(q,d)))$$

Experiment and Result

Parameter Setting

LM

The prior collection sample size μ is set to 2000 across all the experiments which is also used in LLM and PLM.

LLM

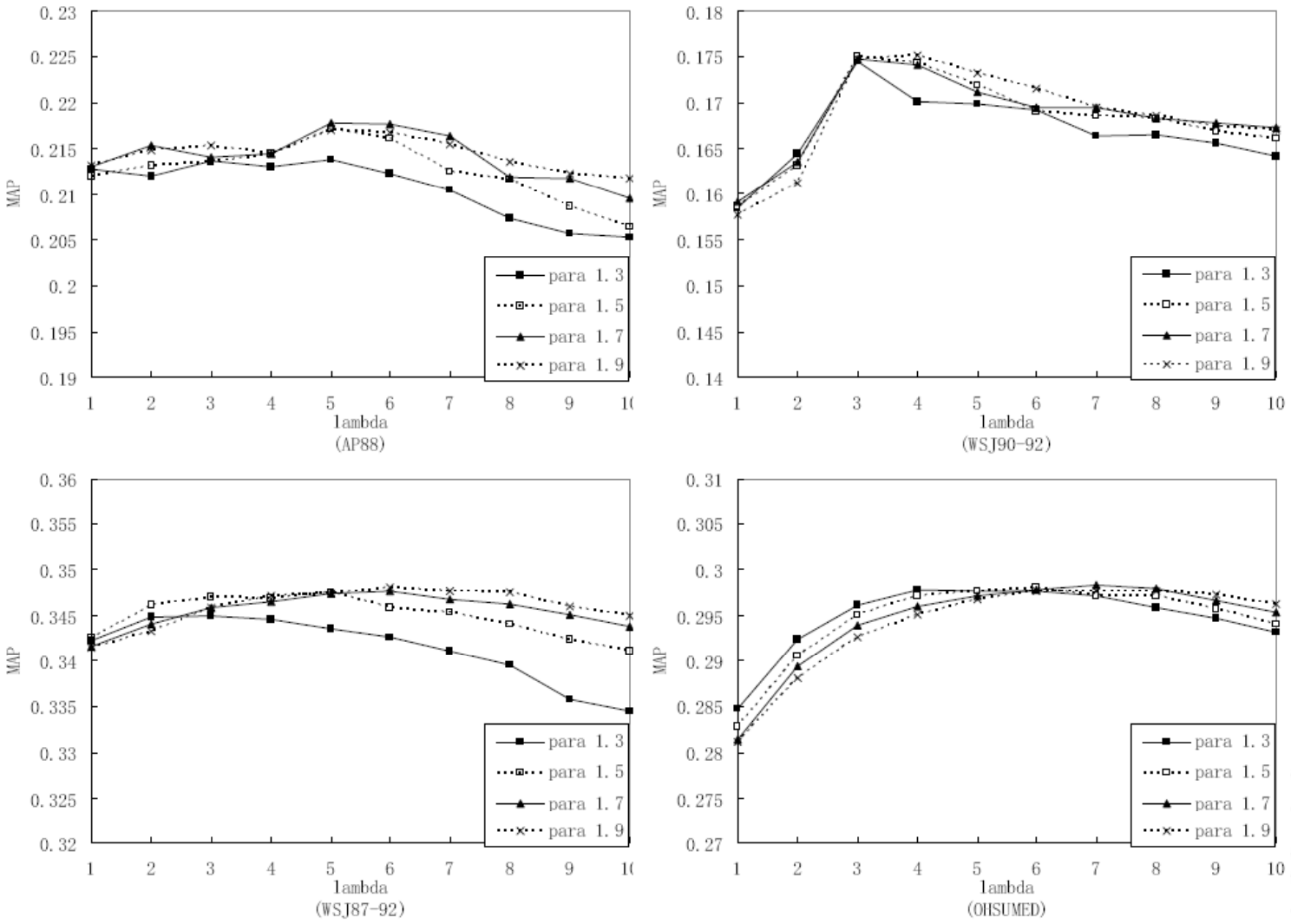
Parameter is optimized by searching : 0.1, 0.2, ..., 1.0.

PLM

- Proximity argument λ:controls the proportional weight of prior proximity factor relative to the observed word count information.
- Exponential weight para:controls the proportional ratio of proximity score between different query terms.
- Optimization space: para : 1.1, 1.2, ..., 2.0, λ : 0.1, 1, 2, 3, ..., 10.

Experiment and Result

PLM's parameter Sensitivity using P_MinDist.



Experiment and Result

Comparison of Best Performance

Model/Measure	MAP	PR@5	PR@10	MAP	PR@5	PR@10
	AP88			WSJ90-92		
LM	0.2070	0.3120	0.2620	0.1534	0.2160	0.1860
LLM	0.2123	0.3280	0.2720	0.1589	0.2280	0.1940
PLM(P_MinDist)	0.2178	0.3440	0.2780	0.1751*	0.2320	0.1940
PLM(P_AveDist)	0.2167	0.3360	0.2740	0.1603	0.2320	0.1900
PLM(P_SumProx)	0.2203	0.3440	0.2840	0.1735*	0.2360*	0.2000*
	WSJ87-92			OHSUMED		
LM	0.3351	0.5440	0.4960	0.2704	0.4889	0.4698
LLM	0.3457	0.5640	0.5120	0.2651	0.4857	0.4587
PLM(P_MinDist)	0.3474	0.5680	0.5100	0.2983*	0.5365*	0.5095*
PLM(P_AveDist)	0.3446	0.5600	0.5060	0.2954*	0.5238*	0.5095*
PLM(P_SumProx)	0.3493	0.5720	0.5120	0.2984*	0.5397*	0.5154*

Experiment and Result

Main Observation

The observations

- PLM performs empirically better than LM and LLM.
- LLM fails on Ohsumed collection (more verbose in queries).
- PLM performs very well on verbose queries.
- For the three proposed term proximity measures used in PLM, P_*SumProx* and P_*MinDist* performs better than P_*AveDist*.

Experiment and Result

The Influence of Stop Word

Considering stop word in query

- A good ranking function should also perform well when stop words are considered.
- Stop word usually has many occurrences, resulting in a great chance to be proximate with other words in the document.
- Make the proximity mechanism at risk to loose its effect.

Test setting

All the queries from TOPIC251-300 that contain at least one word in the used stop word list.

Experiment and Result

The Influence of Stop Word

Model/Measure	MAP	Pr@5	Pr@10
AP88			
LM	0.1537	0.2957	0.2348
LLM	0.1422	0.2609	0.2435
PLM(P_MinDist)	0.1575	0.2870	0.2391
PLM(P_AveDist)	0.1558	0.3030	0.2479
PLM(P_SumProx)	0.1607	0.2957	0.2565
WSJ90-92			
LM	0.1072	0.1652	0.1348
LLM	0.1017	0.1391	0.1217
PLM(P_MinDist)	0.1139	0.1652	0.1391
PLM(P_AveDist)	0.1080	0.1739	0.1348
PLM(P_SumProx)	0.1158	0.1665	0.1522

Observations

- LLM fails when stop word is considered.
- PLM can still improve on the basic language model.
- P_*SumProx* is the best choice of the three term proximity measures.

Underlying Reason

- Stop word affect LLM globally.
- Stop word affect PLM locally.

Conclusions

STS2009 2nd Search Technology Summit

Main Contribution

- Propose a novel way to integrate proximity factor into the unigram language modeling.
  - The model views query terms' proximate centrality as Dirichlet hyper-parameters.
  - A term's score contribution is boosted when it is at a high proximate area among query terms.
- This integration method is mathematical grounded and shows empirical better performance.
- Besides simple keyword query, this model also performs well in verbose query and stop word containing query.
- Illustrate a way to model more beyond BOW under BOW.

Conclusions

Future Work

- Develop a more efficient way to set the parameters of the PLM model instead of using exhaustive search.
- Study how to normalize the term proximity centrality to a given scale or even to probability.
  - See the effect on parameter tuning.
  - See the effect on ranking result.
- Study how to combine the proximity with other document information such as prior document strength to further improve the effectiveness of language modeling.

Thank you!

Any questions?