Improving Information Retrieval Performance using Word Embedding

Dwaipayan Roy

Indian Statistical Institute, Kolkata

December 28, 2018

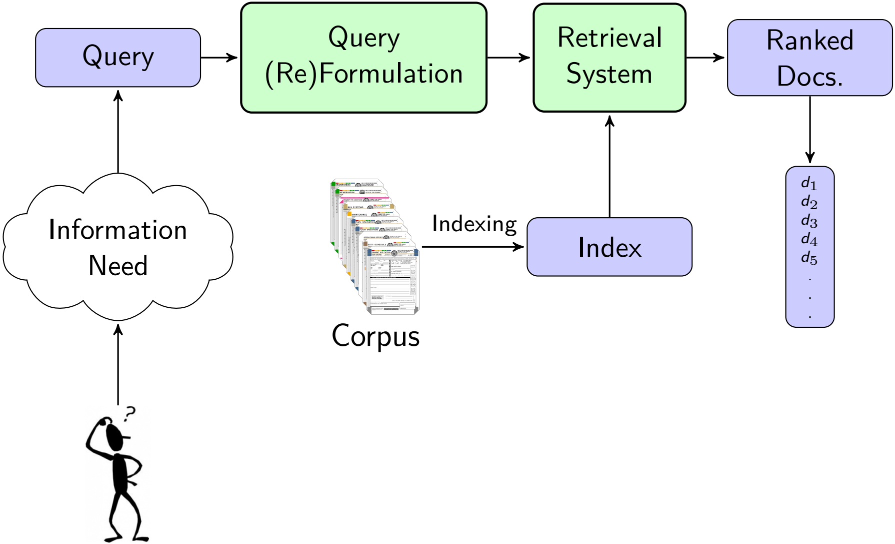
Information Retrieval

www.trec.nist.gov

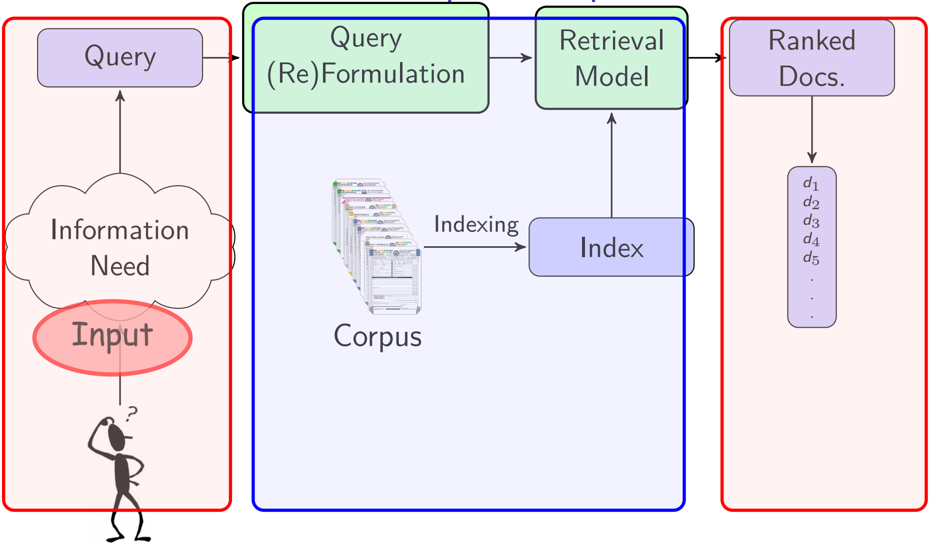
Information Retrieval: A Practical Application

www.google.com

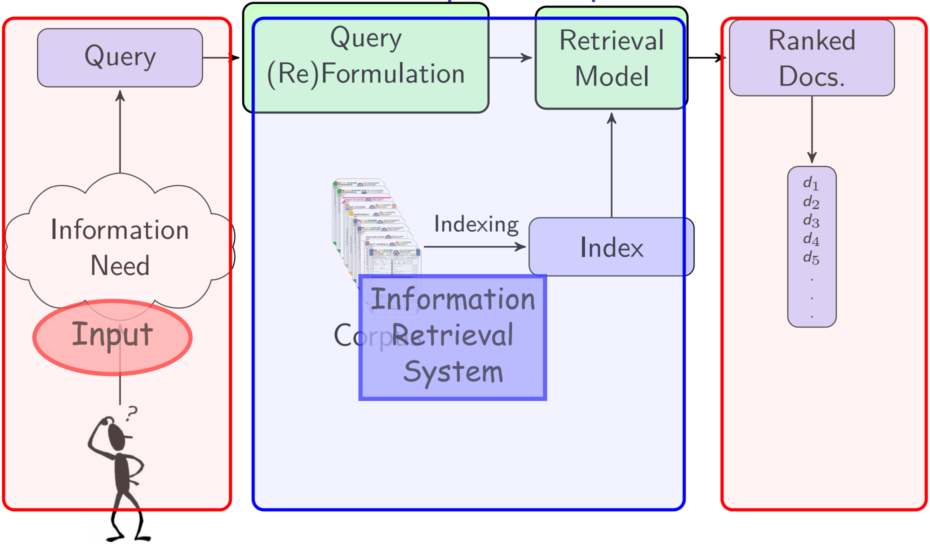
Information Retrieval: A Graphical Representation



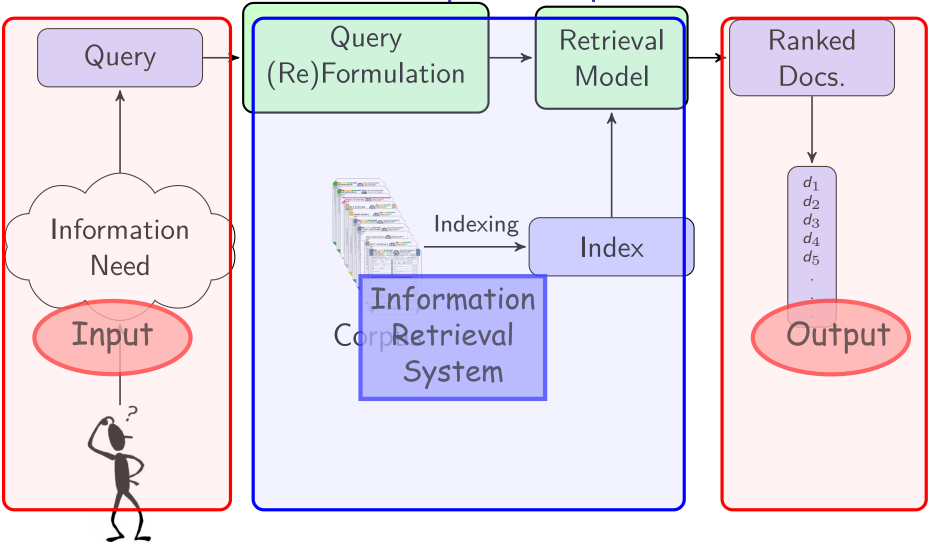
Information Retrieval: A Graphical Representation



Information Retrieval: A Graphical Representation



Information Retrieval: A Graphical Representation



Retrieval Models

- Language Model with Jelinek Mercer Smoothing (LM-JM)
- Language Model with Dirichlet Smoothing (LM-Dir)
- BM25
- DFR
- ...

Retrieval Models

- Language Model with Jelinek Mercer Smoothing (LM-JM)
- Language Model with Dirichlet Smoothing (LM-Dir)
- BM25
- DFR
- ...

$$\text{Score(D,Q)} = \begin{cases} \sum_{t \in Q} f(D,t) \\ \prod_{t \in Q} f(D,t) \\ \dots \end{cases}$$

Retrieval Models

- Language Model with Jelinek Mercer Smoothing (LM-JM)
- Language Model with Dirichlet Smoothing (LM-Dir)
- BM25
- DFR
- ...

$$\text{Score(D,Q)} = \begin{cases} \sum_{t \in Q} f(D,t) \\ \prod_{t \in Q} f(D,t) \\ \ldots \end{cases}$$

$$\text{LM-JM(D,Q)} = \prod_{t \in Q} \Big[\lambda * \text{MLE(t in Document)} + (1-\lambda) * \text{MLE(t in Collection)}\Big]$$

Evaluation: Datasets

Collection Name	Documents	# documents	# topics	# rel-docs
TREC123	Tipster disks 1, 2	741,856	150 (51-200)	37836
TREC678	Tipster disks 4, 5 exclude docs. from CR	528,155	150 (301-450)	13692
Robust	Tipster disks 4, 5 exclude docs. from CR	528,155	100 (601-700)	3720
TREC910	WT10G	1,692,096	100 (451-550)	5980
GOV2	GOV2	25,205,179	150 (701-850)	26917
ClueWeb09B	ClueWeb09 Disk1 - English	50,220,423	200 (1-200)	11037

Table: Overview of datasets used in experiments reported in this presentation.

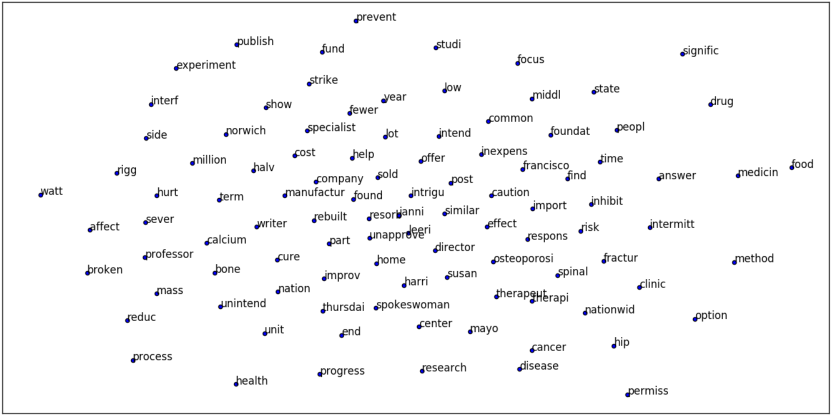
Evaluation: Metrics

- MAP: Mean average precision
- P@5: Precision upto rank 5
- Recall: Percentage of relevant documents retrieved

Word Embedding

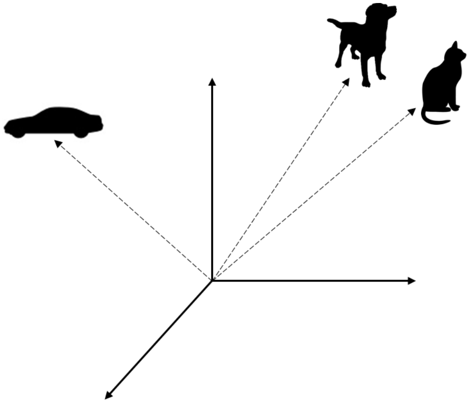
Word Embedding

- Represents every word as low dimensional vector in an abstract space.



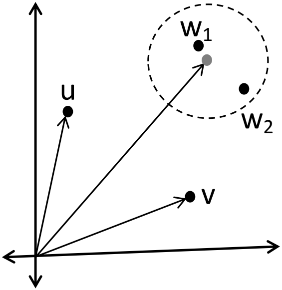
Word Embedding

- Represents every word as low dimensional vector in an abstract space.
- Similarity between vectors reflect semantic similarity between terms.



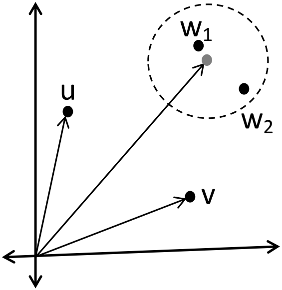
Word Embedding

- Represents every word as low dimensional vector in an abstract space.
- Similarity between vectors reflect semantic similarity between terms.
- Effect of conceptual composition by simple addition of vectors.



Word Embedding

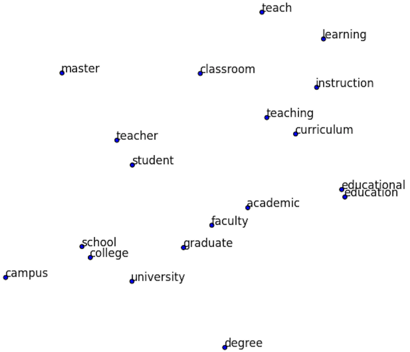
- Represents every word as low dimensional vector in an abstract space.
- Similarity between vectors reflect semantic similarity between terms.
- Effect of conceptual composition by simple addition of vectors.



$$\textbf{German} + \textbf{Airlines} \sim \textbf{Lufthansa}$$

Word Embedding

- Represents every word as low dimensional vector in an abstract space.
- Similarity between vectors reflect semantic similarity between terms.
- Effect of conceptual composition by simple addition of vectors.
- Words have close vector representations if they share similar contexts.



Outline

Outline

1. Improving Baseline Retrieval Model
2. Improving Relevance Feedback based Query Expansion
3. Query Performance Prediction using Word Embedding

Traditional Language Model

Given:

- Collection C: A set of documents
- Query Q

Documents ranked:

- in decreasing order by the posterior probabilities $P(D|Q)$

Traditional Language Model

$$P(D|Q) = \prod_{t \in Q} \lambda \frac{tf(t, D)}{|D|} + (1 - \lambda) \frac{cf(t)}{cs}$$

Query terms generated by independent sampling from either the document or the collection

Language Model with Word Embedding

- A generative process in which a noisy channel may transform a term t' into a term t

Term Transformation Events : via Document Sampling

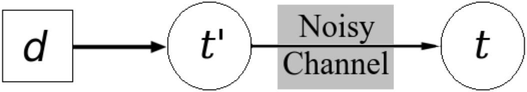


Figure: Schematics of generating a query term t from d

$$P(t, t'|d) = P(t|t', d)P(t'|d)$$
$$P(t|t', d) = \frac{sim(t', t)}{\Sigma(d)}$$

Term Transformation Events : via Collection Sampling

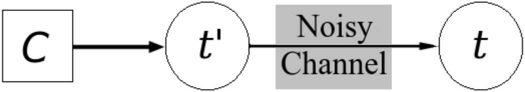


Figure: Schematics of generating a query term t from the collection

$$P(t, t'|C) = P(t|t', C)P(t'|C)$$
$$P(t|t', C) = \frac{sim(t,t')}{\sum_{t'' \in N_t} sim(t,t'')}$$

Term Transformation Events

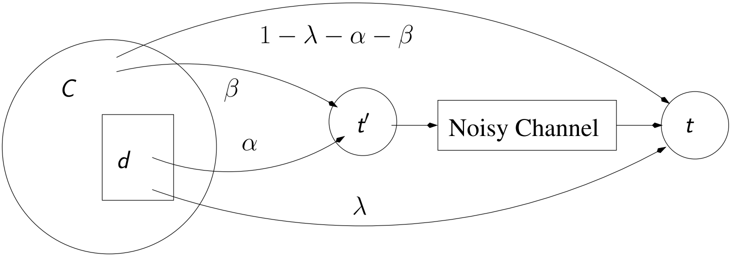


Figure: A Generalized Language Model (GLM). GLM degenerates to LM when $\alpha = \beta = 0$.

- λ**:** Standard Jelinek Mercer weighting parameter
- α**:** Probability of sampling term t via a transformation through a term t' sampled from the document d
- β**:** Probability of sampling term t via a transformation through a term t' sampled from collection

Term Transformation Events : Combined Events

$$P(Q|d) = \prod_{t \in Q} \Big[\lambda P(t|d) + \alpha \sum_{t' \in d} P(t, t'|d) P(t') + \\ \beta \sum_{t' \in N_t} P(t, t'|C) P(t') + (1 - \lambda - \alpha - \beta) P(t|C) \Big]$$

Generalized Language Model : Evaluation

Query	Method	Metrics		
		MAP	P@5	Recall
TREC 123	LM	0.1967	0.4213	0.4477
	GLM	**0.2104**†	**0.4562**†	**0.4893**†
TREC 678	LM	0.2189	0.4160	0.5300
	GLM	**0.2214**†	**0.4187**	**0.5327**
Robust	LM	0.2658	0.4364	0.7881
	GLM	**0.2777**†	**0.4586**†	**0.7990**
WT10G	LM	0.1747	0.3152	0.6377
	GLM	**0.1906**†	**0.3782**†	**0.6689**†

Table: Comparative performance of GLM on the basis of mean average precision (MAP) precision at 5 (P@5) and recall at 1000. A † indicates the significance of the metric value with respect to the baseline LM based retrieval model.

Evaluation: Summary

GLM $>$ LM

Evaluation: Summary

GLM $>$ LM

A Word Embedding based Generalized Language Model for Information Retrieval (SIGIR 2015)

Retrieval Models: Keyword Matching

$$\text{Score(D,Q)} = \begin{cases} \sum_{t \in Q} f(D, t) \\ \prod_{t \in Q} f(D, t) \\ \ldots \end{cases}$$

Retrieval Models: Keyword Matching

$$\text{Score(D,Q)} = \begin{cases} \sum_{t \in Q} f(D, t) \\ \prod_{t \in Q} f(D, t) \\ \ldots \end{cases}$$

Query: *Vehicle Smoke Pollution*

$$\text{Score(D,Q)} = f(D, Vehicle) * f(D, Smoke) * f(D, Pollution)$$

Vocabulary Mismatch

Vehicle
Smoke
Pollution

`Doc1`

Passenger vehicles & heavy-duty trucks are a major source of air pollution which includes ozone, particulate matter and other smog-forming emissions in the form of smoke.

`Doc2`

The exhaust gas of cars powered by fossil fuels are a major source of toxic materials in the air that causes severe damage to the eco-system.

Vocabulary Mismatch

Vehicle
Smoke
Pollution

Doc1

Passenger vehicles & heavy-duty trucks are a major source of air pollution which includes ozone, particulate matter and other smog-forming emissions in the form of smoke.

Doc2

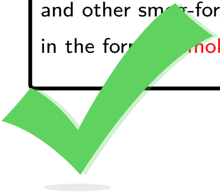
The exhaust gas of cars powered by fossil fuels are a major source of toxic materials in the air that causes severe damage to the eco-system.

Vocabulary Mismatch

Vehicle
Smoke
Pollution

Doc1

Passenger vehicles & heavy-duty trucks are a major source of air pollution which includes ozone, particulate matter and other sm[illegible]-forming emissions in the for[illegible]moke.



Doc2

The exhaust gas of cars powered by fossil fuels are a major source of toxic materials in the air that causes severe damage to the eco-system.

Vocabulary Mismatch: Solution

Query Expansion

Add more words of similar meaning to initial query for better retrieval.

Outline

1. Improving Baseline Retrieval Model
2. Improving Relevance Feedback based Query Expansion
3. Query Performance Prediction using Word Embedding

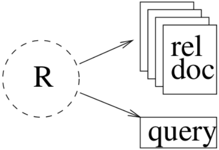
Outline

Pseudo-Relevance Feedback based Query Expansion

Given a query Q

- Perform initial retrieval using some retrieval model
- Consider top K documents as (pseudo-)relevant
- Select terms from those documents as candidate expansion terms
- Perform re-retrieval with the expanded query

Relevance based Language Model (RLM)



- Assumes that both query and (pseudo-)relevant documents sampled from a latent relevance model $\mathcal{R}$.
- The task $\rightarrow$ to find (estimate) the density function for $\mathcal{R}$.

$$P(w|R) = \sum_{D \in M} P(w|D) \prod_{q \in Q} P(q|D)$$

RM3

RM3

A mixture model of RLM and query likelihood model.

$$P'(w|R) = \mu P(w|R) + (1-\mu) P(w|Q)$$

$$P(w|R) = \sum_{D \in M} P(w|D) \prod_{q \in Q} P(q|D)$$

$$P(w|Q) = \frac{tf(w,Q)}{|Q|}$$

Motivation of this work

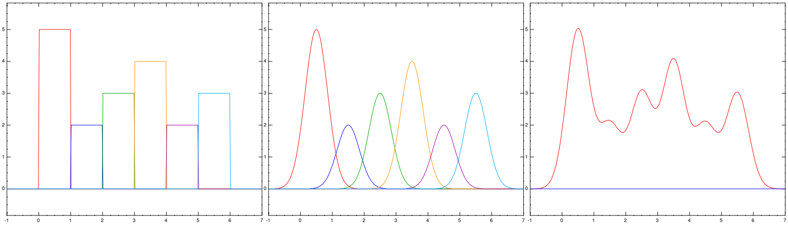
Word Vector Compositionality based Relevance Feedback using Kernel Density Estimation

Motivation of this work

Word Vector Compositionality based Relevance Feedback using Kernel Density Estimation

- **Word embedding:**
  - Captures the semantic relationship between terms.
- **Relevance feedback:**
  - Captures co-occurence information of terms with query in top docs.

Kernel Density Estimation (KDE)



- Estimate a distribution that generates the given data (data points).
- Place kernel function (e.g. Gaussian) centered around each observed data point.
- Combine the Gaussians to get a function peaked at the observed data point.

Kernel Density Estimation (KDE)

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

- h - bandwidth (smoothing parameter)
- n - number of data points
- x_i - i^{th} data point
- $K()$ - the kernel

KDE with Gaussian Kernel

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

KDE with Gaussian Kernel

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(\frac{x-x_i}{h})^2}$$

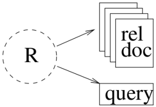
KDE with Gaussian Kernel

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(\frac{x-x_i}{h})^2}$$

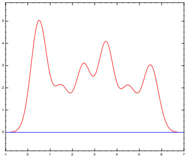
- h and σ: tunable parameters

Comparing the Estimations



Relevance Model Estimation

Given set of terms from (pseudo-)relevant documents, estimate the density function that generates $\mathcal{R}$.



Kernel Density Estimation

Given set of n data samples, estimates the density function that generates the observed data.

KDE in Relevance Feedback Scenario

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

The Points

- **Observed data points (x_i):** $Q = \{q_1, \ldots, q_n\}$
- **Points at which probability (density) is to be estimated (x):** $w \rightarrow$ Candidate expansion term

KDE in Relevance Feedback Scenario

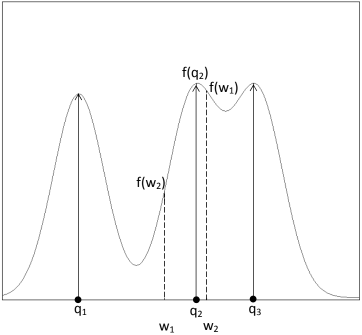
$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

The Points

- **Observed data points (x_i):** $Q = \{q_1, \ldots, q_n\}$
- **Points at which probability (density) is to be estimated (x):** $w \rightarrow$ Candidate expansion term

$$f(w) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{w - q_i}{h})$$

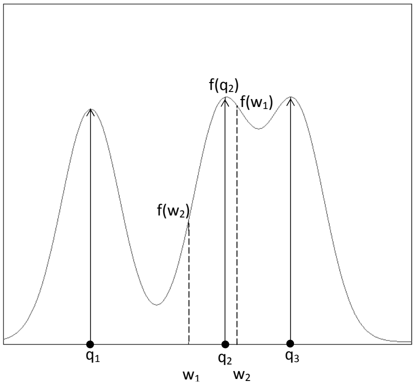
Relevance Feedback with KDE



- One dimensional projection of embedded vectors

Kernel Density Estimate (Unweighted)

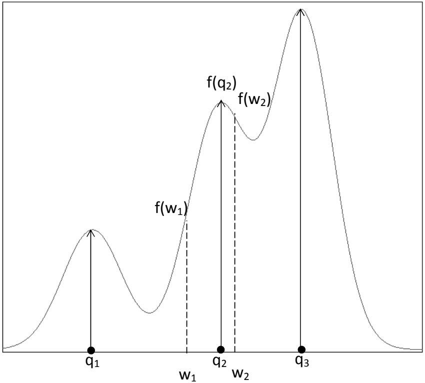
$$f(w) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{w - q_i}{h})$$



- n - number of samples
- h - bandwidth
- $K(\cdot)$ - kernel function

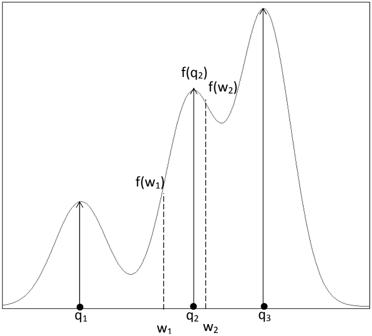
Kernel Density Estimate (Weighted)

$$f(w, \alpha) = \frac{1}{nh} \sum_{i=1}^{n} \alpha_i K(\frac{w - q_i}{h})$$



- n - number of samples
- h - bandwidth
- $K(\cdot)$ - kernel function
- α_i - weight of local Kernel function around i^{th} data point

One dimensional KDE



$$f(w, \alpha) = \frac{1}{nh} \sum_{i=1}^{n} \alpha_i \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(w-q_i)^2}{2\sigma^2 h^2}}$$

- α_i : weight of the i^{th} query term

One dimensional KDE

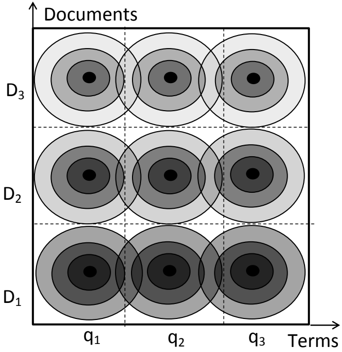
$$f(w, \alpha) = \sum_{i=1}^{n} \boxed{P(w|\mathcal{M})P(q_i|\mathcal{M})} \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{\boxed{(\mathbf{w} - \mathbf{q_i})^T(\mathbf{w} - \mathbf{q_i})}}{2\sigma^2 h^2})$$

- Captures co-occurence information of terms with query in top docs.
- Captures the semantic relationship between terms.

Two dimensional KDE

- In 1D-KDE, set of all top ranked documents considered as a single document model ($\mathcal{M}$).
- We now generalize the 1D-KDE to two dimensions so as to weight the contributions obtained from different documents in the ranked list separately.

Two dimensional KDE



- The x-axis corresponds to the query term vectors similar to one dimensional KDE (unordered).
- The y-axis represents the normalized term frequency of query terms in respective feedback documents.

Two dimensional KDE

The Points: One Dimensional KDE

- **Observed data points (x_i):** $Q = \{q_1, \ldots, q_n\}$
- **Points at which probability (density) is to be estimated (x):**
  $w \rightarrow$ Candidate expansion term

Two dimensional KDE

The Points: Two Dimensional KDE

- **Observed data points ($\mathbf{x_{ij}}$)** $= (q_i, D_j)$: Encapsulates word vector for i-th query word q_i and its normalized term frequency in feedback document D_j ($P(q_i|D_j)$).
- **Points at which probability (density) is to be estimated ($\mathbf{x}$)** $= (w, D_j)$: Encapsulates word vector for candidate expansion term w and its normalized term frequency in feedback document D_j ($P(w|D_j)$).

Two dimensional KDE

$$f(\mathbf{x}, \alpha) = \sum_{i=1}^{n} \sum_{j=1}^{M} \frac{P(w|D_j) P(q_i|D_j)}{2\pi\sigma^2} \exp\left(\frac{(\mathbf{w} - \mathbf{q_i})^2 + (P(w|D_j) - P(q_i|D_j))^2}{-2\sigma^2 h^2}\right)$$

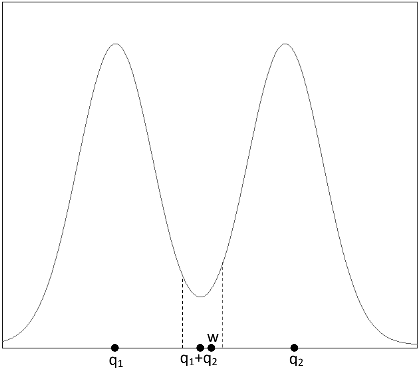
Two dimensional KDE

$$f(\mathbf{x}, \alpha) = \sum_{i=1}^{n} \sum_{j=1}^{M} \frac{P(w|D_j)P(q_i|D_j)}{2\pi\sigma^2} \exp\left(\frac{(\mathbf{w} - \mathbf{q_i})^2 + (P(w|D_j) - P(q_i|D_j))^2}{-2\sigma^2 h^2}\right)$$

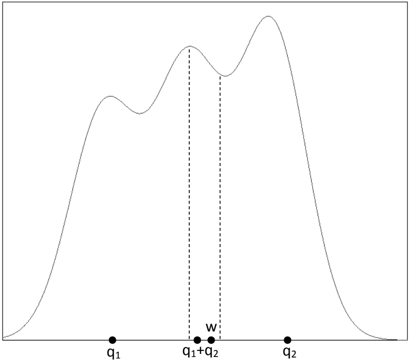
- w in document D_j, denoted as $\mathbf{x}$, will get a high value of the density function if:
  - (i) $\mathbf{w}$ is semanticlly close to query terms $\mathbf{q_i}$;
  - (ii) w frequently co-occurs with query terms in each top ranked document D_j.

Term Composition in KDE

- Extending the set of pivot points:

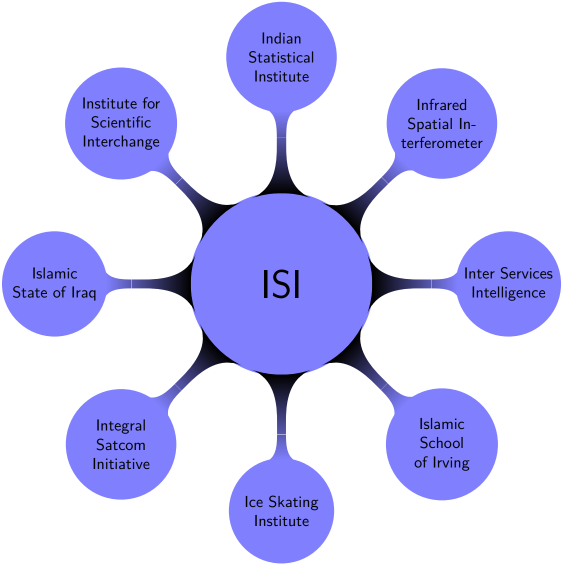


(a) Without composition

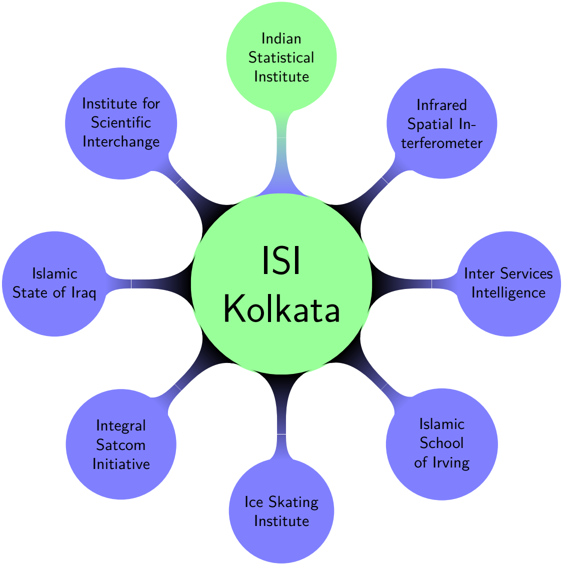


(b) With composition

Importance of Composition



Importance of Composition



Summary

Given:

- Corpora : Set of documents.
- Query Q : $q_1, \ldots, q_n$.
- Vector embeddings of each of the terms.

- First level retrieval performed using any LM.
- ET = All terms of top ranked M documents considered as potential expansion terms.
- Calculate KDE(w) for each $w \in ET$.
- Terms with high estimated value taken as expansion terms.
- Linearly interpolate derived density function with underlying query model.

Evaluation

Query	Method	LM-JM		
		MAP	P@5	Recall
TREC 123	LM	0.1967	0.4213	0.4477
	RM3	0.2507*	0.4653*	0.5154*
	KDERLM	**0.2661***†	**0.4798***	**0.5373***†
TREC 678	LM	0.2189	0.4160	0.5300
	RM3	0.2351*	0.4547*	0.5366
	KDERLM	**0.2536***†	**0.4601***†	**0.5520***†
Robust	LM	0.2658	0.4364	0.7881
	RM3	0.3309*	0.4929*	**0.8596***
	KDERLM	**0.3420***†	**0.5043***	**0.8596***
WT10G	LM	0.1747	0.3152	0.6377
	RM3	0.2094*	0.3394*	0.6743*
	KDERLM	**0.2221***†	**0.3419***	**0.6910***†

Evaluation: Summary

$$\text{KDERLM} > \text{RM3} > \text{GLM} > \text{LM}$$

Evaluation: Summary

KDERLM $>$ RM3 $>$ GLM $>$ LM

Word Vector Compositionality based Relevance Feedback using Kernel Density Estimation (CIKM 2016)

Query Expansion for Simple Query

Searching data set

- National Geographic Readers: Snakes!

Query Expansion for Simple Query

Searching data set

- National Geographic Readers: Snakes!

Query

- python life span

Query Expansion for Simple Query

Searching data set

- National Geographic Readers: Snakes!

Query

- python life span
- python setup in windows 10

Query Expansion for Simple Query

Searching data set

- National Geographic Readers: Snakes!

Query

- python life span
- python setup in windows 10

Query expansion $\rightarrow$ overkill or, not work at all.

Query Expansion for Simple Query

Searching data set

- National Geographic Readers: Snakes!

Query

- python life span
- python setup in windows 10

Query expansion $\rightarrow$ overkill or, not work at all.
How to understand whether query actually need expansion? $\rightarrow$ QPP

Outline

Query Performance Prediction (QPP)

Definition

To quantify the quality of search results when no relevance feedback is given.

- For an ambiguous query, it may be difficult for a search engine to return satisfactory results for the query.
- This can lead to poor IR effectiveness.

Query Performance Prediction (QPP)

Why we need QP Predictor?

- Feedback to Users.
- Feedback to Search Engine.
- Feedback to System Administrator.

What Makes a Query Difficult?

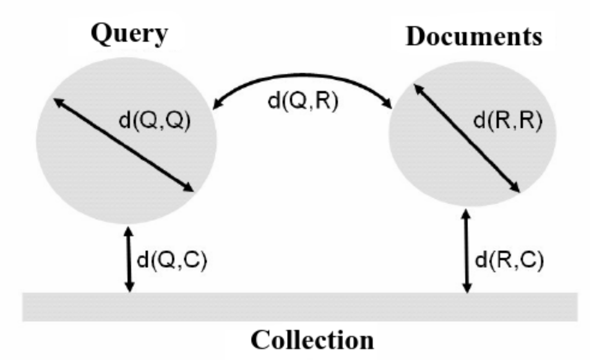
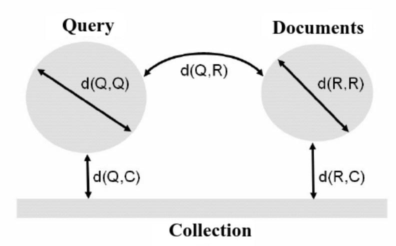


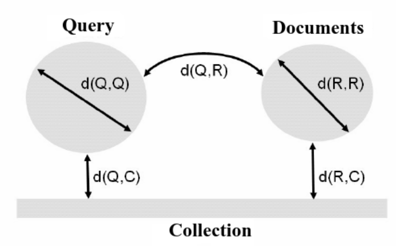
Figure: A general model of a topic based on a query Q expressing a specific information need, the relevant documents R for Q, the entire collection C, and the distances between the sets involved.

Characteristics of Difficult Query



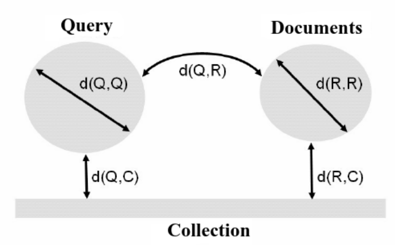
- $d(Q, C)$ - Distance between query and collection; Reflects likelihood of generation of Q from C.

Characteristics of Difficult Query



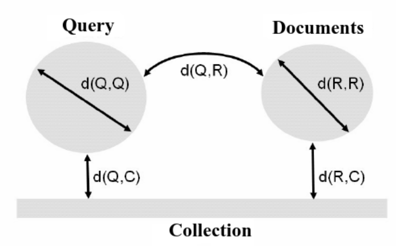
- $d(Q, C)$ - Distance between query and collection; Reflects likelihood of generation of Q from C.
- $d(R, C)$ - Distance between retrieved documents and collection; Reflects likelihood of generation of documents of R from C.

Characteristics of Difficult Query



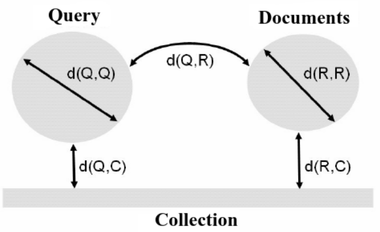
- $d(Q, C)$ - Distance between query and collection; Reflects likelihood of generation of Q from C.
- $d(R, C)$ - Distance between retrieved documents and collection; Reflects likelihood of generation of documents of R from C.
- $d(Q, Q)$ - Diversity of query. Reflects the ambiguity of Q; more the diameter, more ambiguous the query.

Characteristics of Difficult Query



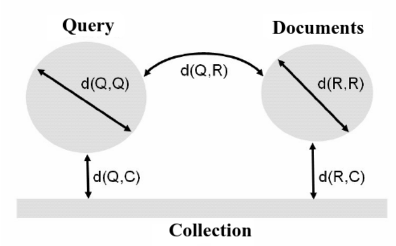
- $d(Q, C)$ - Distance between query and collection; Reflects likelihood of generation of Q from C.
- $d(R, C)$ - Distance between retrieved documents and collection; Reflects likelihood of generation of documents of R from C.
- $d(Q, Q)$ - Diversity of query. Reflects the ambiguity of Q; more the diameter, more ambiguous the query.
- $d(R, R)$ - Distance among retrieved documents; Manifestation of ambiguity of retrieved documents.

Characteristics of Difficult Query



- $d(Q, C)$ - Distance between query and collection; Reflects likelihood of generation of Q from C.
- $d(R, C)$ - Distance between retrieved documents and collection; Reflects likelihood of generation of documents of R from C.
- $d(Q, Q)$ - Diversity of query. Reflects the ambiguity of Q; more the diameter, more ambiguous the query.
- $d(R, R)$ - Distance among retrieved documents; Manifestation of ambiguity of retrieved documents.
- $d(Q, R)$ - Distance between query and retrieved documents R; Equivalent to distribution of retrieval scores of R.

Characteristics of Difficult Query



- $d(Q, C)$ - Distance between query and collection; Reflects likelihood of generation of Q from C.

- $d(Q, Q)$ - Diversity of query. Reflects the ambiguity of Q; more the diameter, more ambiguous the query.

Word Embedding based QPP

Ambiguous Terms

Terms with multiple senses

Hypothesis

For an ambiguous term w, in terms of the embedded space, w is more likely to be a peripheral point of a word cluster rather than being an interior point close to the cluster centre.

Ambiguous Term in Embedded Space

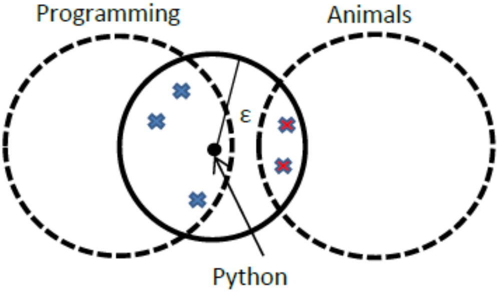


Figure: The neighbourhood of an ambiguous word with multiple senses

Ambiguous Term in Embedded Space

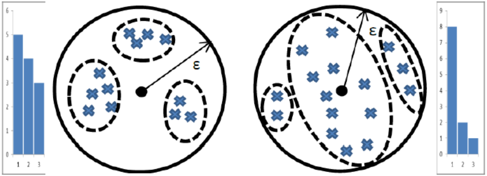


Figure: Histogram of number of terms in the ϵ-neighbourhood of a term

Ambiguous Term in Embedded Space

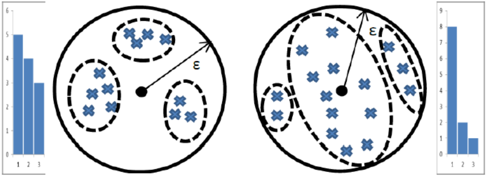


Figure: Histogram of number of terms in the ϵ-neighbourhood of a term

- Cluster the set of points in the ϵ-neighbourhood of query term
- For an ambiguous word, the distribution of points in the clusters likely to be close to uniform

Word Embedding based QPP

$$AMB_{WV}(Q; K, \epsilon) = \frac{1}{K} \sum_{k=1}^{K} (|N_\epsilon(\mathbf{Q})| - \mu)^2;$$

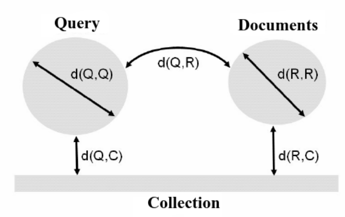
- Q - The query.
- $N_\epsilon(\mathbf{Q})$ - ϵ-neighborhood of $\mathbf{Q}$.
- K - Number of clusters.
- ϵ - Neighbourhood size.
- μ - Average cardinality of clusters. $(\frac{|N_\epsilon(\mathbf{Q})|}{K})$

Word Embedding based QPP

$$AMB_{WV}(Q; K, \epsilon) = \frac{1}{K} \sum_{k=1}^{K} (|N_\epsilon(\mathbf{Q})| - \mu)^2;$$

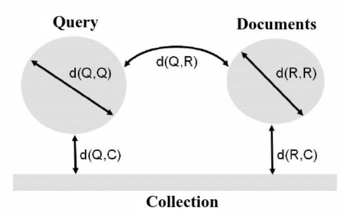
- High variance $\rightarrow$ One sense predominates others $\rightarrow$ **Non-ambiguous**
- Low variance $\rightarrow$ Having multiple senses $\rightarrow$ **Ambiguous**

Pre-retrieval QPP



- $d(Q, C)$ - Reflects likelihood of generation of Q from C.
- $d(Q, Q)$ - Reflects the ambiguity of Q;

Pre-retrieval QPP



- $d(Q, C)$ - Reflects likelihood of generation of Q from C.
- $d(Q, Q)$ - Reflects the ambiguity of Q;

Pre-retrieval QPP Methods

- Ignores any post-retrieval information
- Hybrid of pre-retrieval and post-retrieval approaches outperforms both

Combining with NQC

NQC

- Post-retrieval QPP.
- Variance of similarity score proportional to query performance.

Word Embedding and NQC based Hybrid Predictor

$$AMB_{WV}\text{-NQC} = \alpha AMB_{WV} + (1-\alpha) NQC$$

Evaluation: Metrics

- Correlation between QPP scores and true score (AP value)

Evaluation Metrics

- Pearson's ρ: Correlation coefficient
- Kendall's τ: Rank correlation coefficient

High correlation $\rightarrow$ Better prediction

Results on Some of the Topic sets

	TREC3		TREC7		TREC8		TREC10	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ
AvgIDF	0.2353	0.2490	0.4262	0.3796	0.5910	0.3518	0.3736	0.2359
MaxIDF	0.2285	0.2772	0.3524	0.2662	0.4938	0.2996	0.2219	0.1233
WordNet	0.0297	-0.0375	0.2696	0.2509	0.0746	0.1014	0.2221	0.0896
AMB_{WV}	0.3222	0.2457	0.2043	0.2132	0.2132	0.1559	0.1544	0.0033
NQC	0.3146	0.1559	0.4580	0.3502	0.6717	0.4335	0.4676	0.2686
AvgIDF-NQC	0.2782	0.2620	0.4551	**0.4253**	0.6315	0.3878	0.4119	0.2588
MaxIDF-NQC	0.2862	0.3159	0.4204	0.3094	0.5815	0.3714	0.2157	0.2336
WordNet-NQC	0.1794	0.1069	0.4095	0.3453	0.3457	0.3078	0.3292	0.1673
AMB_{WV}-NQC	**0.4741**	**0.3648**	**0.5362**	0.3844	**0.7070**	**0.4498**	**0.4936**	**0.2914**

Table: Comaprisons of the word embedding based QPP method against various baselines on the test topic sets.

Contribution

- An embedding based approach to quantify underlying ambiguity of user query.

Contribution

- An embedding based approach to quantify underlying ambiguity of user query.

Estimating Gaussian Mixture Models in the Local Neighbourhood of Embedded Word Vectors for Query Performance Prediction
(Information Processing and Management - in press)

Conclusion

Conclusion

WE & **I R** Great together!

Conclusion

WE & **I** **R** Great together!

https://github.com/dwaipayanroy
dwaipayan_r@isical.ac.in

THANK YOU!

GLM: A less-heavy QE method

GLM Parameter Details

- λ empirically set to 0.2
- GLM parameters α and β varied within range of $[0.1, 0.4]$ to ensure $\alpha + \beta + \lambda < 1$
- Word vectors embedded in a 200-dimensional space with negative-sampling using 5-word window on continuous bag-of- words model.
- N_t set to 3

Back

1D-KDE Details

- $P(w|\mathcal{M})$: $\frac{tf(w,\mathcal{M})}{|\mathcal{M}|}$
- $P(q_i|\mathcal{M})$: $\frac{tf(q_i,\mathcal{M})}{|\mathcal{M}|}$
- h: Bandwidth set to one (1)
- σ: Trained on the development set (TREC 6 and TREC 9)

Return

2D-KDE Details

- $(w - q_i)^2 = (\mathbf{w} - \mathbf{q_i})^T(\mathbf{w} - \mathbf{q_i})$
- $P(w|D_j) : \frac{tf(w,D_j)}{|D_j|}$
- $P(q_i|D_j) : \frac{tf(q_i,D_j)}{|D_j|}$
- h: Bandwidth set to one (1)
- σ: Trained on the development set (TREC 6 and TREC 9)
- covariance matrix σ as a diagonal matrix with equal covariance for both the dimensions.

Return

Extra slide

◂ Return

Pearson's ρ

$$\rho_{(X,Y)} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- Range: $(-1, 1)$

Back

Kendall's τ

$$\tau_{(X,Y)} = \frac{\text{(No. of concordant pair) - (No. of discordant pair)}}{n(n-1)/2}$$

- Range: $(-1, 1)$

Back