Word Embeddings for IR

Debasis Ganguly

ADAPT Centre,
School of Computing, Dublin City University
Dublin 9, Ireland.

September 26, 2016

Overview

What is Word Embedding?

- Represent every word as a vector in some *abstract* space.
- What are the characteristics of this space?
    - Two terms t_1 and t_2 are *close* if and only if they share similar contexts.
    - *Paris* is close to *France*. Why?
    - If *Paris* is close to *France*, then *Berlin* will be close to *Germany*. Why?

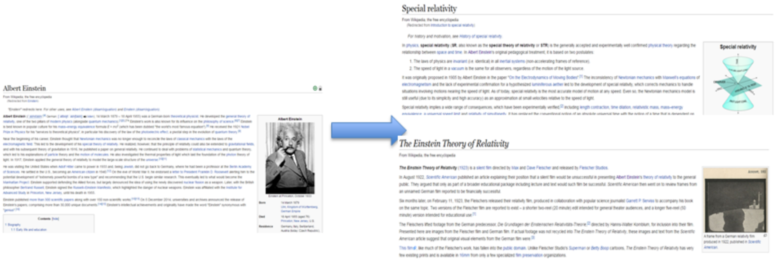
Word Embeddings for Initial Retrieval

- Limitations:
  - Term association: Has been an intriguing problem in IR.
  - Vocabulary mismatch: Different terms may be used in two documents that are about the same topic, e.g. “atomic” and “nuclear” etc.
  - Terms used in query are different from those in its relevant documents.
  - Standard retrieval models assume term independence.
- Proposed Solution:
  - Generalized Language Model, which includes the term transformation in the sampling process by using distances between embedded vectors.

Word Embeddings for Relevance Feedback (RF)

- Limitations:
  - Use statistical co-occurrence of words in top ranked docs with query terms.
  - No way to take into account multi-word 'concepts', e.g. relating 'osteoporosis' to 'bone disease' beyond pre-defined phrases.
  - Noisy expansion terms can lead to 'query drift' and hence degraded IR effectiveness after RF.
- Proposed Solution:
  - Semantic similarity captured by distance measure between word vectors.
  - Integrate semantic similarity with statistical co-occurrences between terms for RF.
  - Exploit term compositionality to extract meaningful concepts to use in RF.

Word Embeddings for Multi-modal IR



Word Embeddings for Multi-modal IR

- Multi-modality: A document comprised of text, images, speech, video, e.g. a typical Wikipage.
- Given a unimodal (e.g. text/image) query or more generally a multi-modal query, how can one retrieve relevant multi-modal documents?
- Standard approach:
  - Index the different modalities separately. Compute similarities individually and fuse.
  - Problems: Different retrieval strategies. How to combine the scores?
- Vector Embedding Approach: Joint embedding of categorical data, such as text, and continuous data such as image features into vectors of reals.
- What we need: A similarity function between sets of vectors.

A Generalized Language Model

- Takes into account term *tansformations* in the sampling method.
- Two types of term transformations (let t be an observed query term):
  - **Document Sampling:** Pick a term t' from d and then change it to t.
  - **Collection Sampling:** Pick a term t' from collection and then change it to t.
- Document sampling transformation measures how well does a term t contextually fits within a document.
- Sampling from collection aims to alleviate vocabulary mismatch.

A schematic diagram

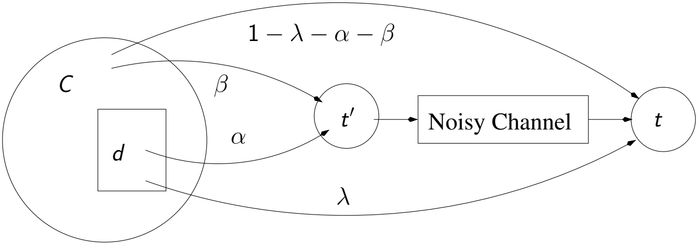


Figure: Schematics of generating a query term t in our proposed Generalized Language Model (GLM). GLM degenerates to LM when $\alpha = \beta = 0$.

Dataset

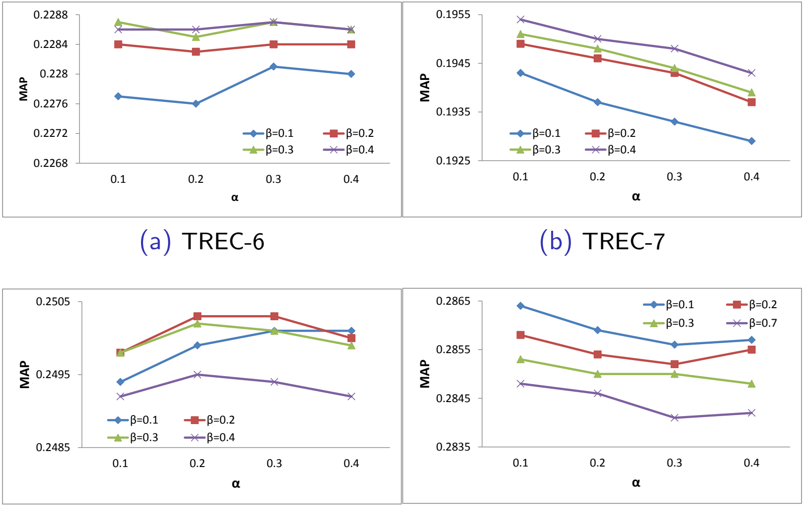
Table: Dataset Overview

Document Collection	Document Type	#Docs	Vocab Size	Query Fields	Query Set	Query Ids	Avg qry length	Avg # rel docs	Dev Set	Test Set
TREC Disks 4, 5	News	528,155	242,036	Title	TREC 6 ad-hoc	301-350	2.48	92.2	✓	
					TREC 7 ad-hoc	351-400	2.42	93.4		✓
					TREC 8 ad-hoc	401-450	2.38	94.5		✓
					TREC Robust	601-700	2.88	37.2		✓
WT10G	Web pages	1,692,096	1,659,231	Title	TREC 9 Web	451-500	3.46	52.3	✓	
					TREC 10 Web	501-550	4.62	67.2		✓

Results

Topic Set	Method	Metrics MAP	GMAP	Recall
TREC-6	LM	0.2148	0.0761	0.4778
	LDA	0.2192	0.0790	**0.5333**
	GLM	**0.2287**	**0.0956**	0.5020
TREC-7	LM	0.1771	0.0706	0.4867
	LDA	0.1631	0.0693	0.4854
	GLM	**0.1958**	**0.0867**	**0.5021**
TREC-8	LM	0.2357	0.1316	0.5895
	LDA	0.2428	0.1471	0.5833
	GLM	**0.2503**	**0.1492**	**0.6246**
Robust	LM	0.2555	0.1290	0.7715
	LDA	0.2623	**0.1712**	**0.8005**
	GLM	**0.2864**	0.1656	0.7967

Parameter Variation Effects



(a) TREC-6 (b) TREC-7

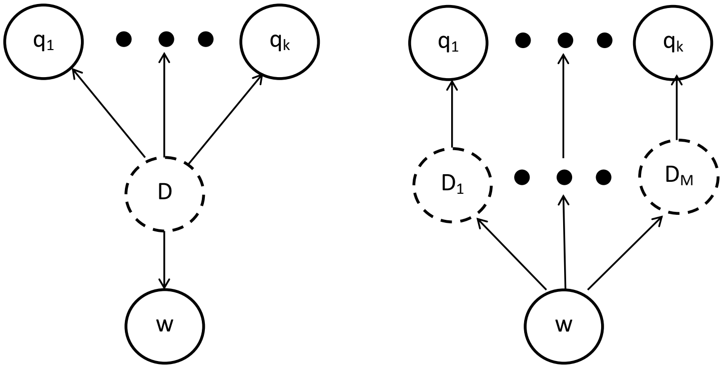
(c) TREC-8 (d) Robust

Figure: GLM parameters' (α and β) effect on MAP.

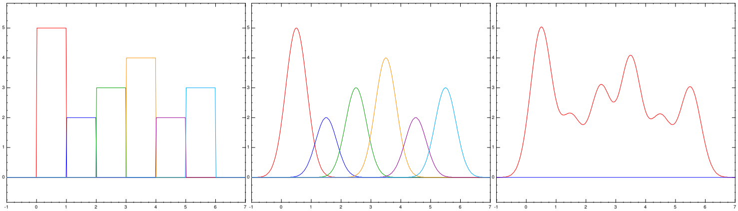
Relevance Model

- Standard approach to relevance feedback with a generative model.
- Estimates a distribution $P(w|Q)$, where w is a term in the set of top docs and Q is the set of query terms.
- Two versions of generative model.
  - **iid**: Terms generated from the whole set of top documents.
  - **conditional**: Terms generated from individual top documents with prior probabilities.

Two variants of the Relevance Model



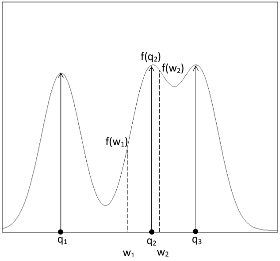
Kernel Density Estimation



- Estimate a distribution that generates the given data.
- Place Gaussians centered around the data points.
- Combine the Gaussians to get a function peaked at the data points.

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

One dimensional KDE



- Query **vector embedded words** are the data points.
- Objective: Estimate the probability distribution function $P(w)$ given the query terms (word vectors).
- High in the neighborhood (of $\mathbb{R}^p$) around query wvecs $\rightarrow$ high $P(w)$ values for terms semantically related to query.
- Low away from neighborhood around query wvecs $\rightarrow$ Terms, semantically unrelated to the query terms, have low $P(w)$.

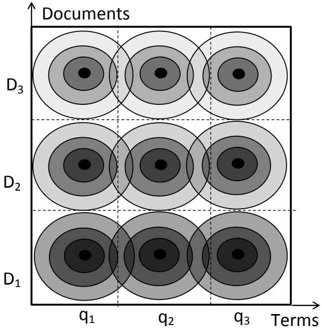
One dimensional KDE (Weighted)

- Put a weight α_i as a coefficient for each kernel function centered around a data point.
- Define $\alpha_i = P(w|D)P(q_i|D)$.
- Define kernel: $K(\frac{w-q_i}{h}) = \mathcal{N}(\frac{w}{h}, \frac{q_i}{h}, \sigma)$.
- Acts as generalized RLM (iid)

$$f(w, \alpha) = \frac{1}{k}\sum_{i=1}^{k} \alpha_i K(\frac{w-q_i}{h}) = \sum_{i=1}^{k} \alpha_i \mathcal{N}(\frac{w}{h}, \frac{q_i}{h}, \sigma)$$

$$= \sum_{i=1}^{k} P(w|D)P(q_i|D)\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(w-q_i)^T(w-q_i)}{2\sigma^2 h^2})$$

Two dimensional KDE



- Word vectors of query terms is one dimension.
- The second dimension is the rank (or similarity) of the documents.
- Objectives: More contribution from:
  - terms that are *closer* to query terms.
  - documents that are ranked higher.

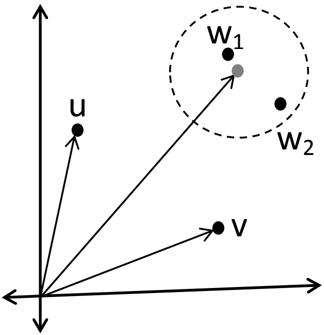
Two dimensional KDE

- Choose kernels as bivariate Gaussians: $K(\frac{w-q_i}{h}) = \mathcal{N}(\frac{w}{h}, \frac{q_i}{h}, \sigma)$.
- Data points: $\mathbf{x_{ij}} = (q_i, D_j)$.
- Put a weight α_i as a coefficient for each kernel function centered around a data point.
- Define $\alpha_{ij} = P(w|D_j)P(q_i|D_j)$.
- Acts as generalized RLM (conditional).

$$f(\mathbf{x}, \alpha) = \sum_{i=1}^{k} \sum_{j=1}^{M} \frac{P(w|D_j)P(q_i|D_j)}{2\pi\sigma^2}$$

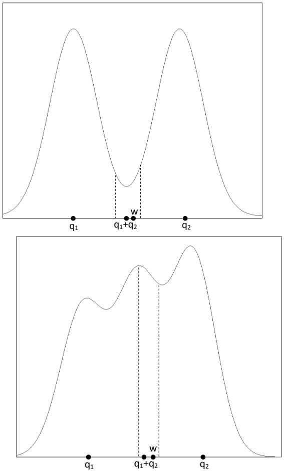
$$\exp(\frac{(w-q_i)^2 + (P(w|D_m) - P(q_i|D_j))^2}{-2\sigma^2 h^2})$$

Vector Addition for Compositionality (Motivation)



- Composition of two (or more) words can lead to a different concept.
- Terms *German* and *airlines* may have high co-occurrence scores with query terms.
- Does not necessarily mean that *Lufthansa* will get a high score.

Composition in KDE Models



- Add a composed point as a pivot point.
- Note how the shape of the function can change.
- Terms (e.g. *Lufthansa*) that are close to the concept of the composed terms get high likelihood.

Parameter tuning on the TREC-6 development set

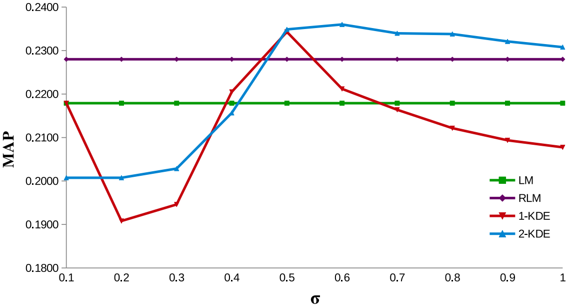


Figure: Effect of varying σ (h fixed to 1) for KDE feedback models on the TREC 6 dataset.

Results on TREC ad-hoc task

Dataset	Method	wvec cmpos	MAP	GMAP	P@5
			Metrics		
TREC 6	LM	-	0.2179	0.0839	0.4040
	RLM	-	0.2280*	0.0871*	**0.4680***‡
	1d KDE	no	0.2307*	0.0842*	0.4359*
	1d KDE	yes	0.2349*	0.0872*	0.4239
	2d KDE	no	0.2369*†	0.0866*	0.4199
	2d KDE	yes	**0.2407***†‡	**0.0908***†‡	0.4640*‡
TREC 7	LM	-	0.1787	0.0830	0.4040
	RLM	-	0.1953*	0.0908*	0.4160*
	1d KDE	no	0.2012*	0.0913*	0.4239*
	1d KDE	yes	0.2107*	0.0938*	0.4440*†
	2d KDE	no	0.2109*†	0.0935*	0.4479*†
	2d KDE	yes	**0.2124***†‡	**0.0943***	**0.4520***†‡
TREC 8	LM	-	0.2466	0.1386	0.4560
	RLM	-	0.2445	0.1448	0.5079
	1d KDE	no	0.2420	0.1510	0.5160
	1d KDE	yes	0.2599	0.1539	**0.5240**
	2d KDE	no	0.2648*†	0.1583	**0.5240**
	2d KDE	yes	**0.2741***†‡	**0.1594***†	0.5120
TREC Robust	LM	-	0.2699	0.1723	0.4464
	RLM	-	0.3105*	0.1956*	0.4989*
	1d KDE	no	0.2932	0.1766	0.4808*
	1d KDE	yes	0.3042	0.1847	0.4869*
	2d KDE	no	0.3158*	0.2015*	**0.5192***†‡
	2d KDE	yes	**0.3327***†‡	**0.2128***†‡	0.5071*

Table: Comparison between KDE and the RLM without QE. Parameters are tuned on the TREC 6 topic set.

Parameter tuning on the TREC-9 development set

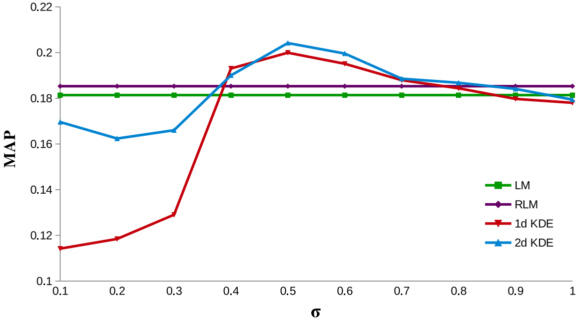


Figure: Effect of varying σ (h set to 1) for KDE feedback models on the TREC 9 topic set.

Results on TREC Web task

Dataset	Method	wvec cmpos	MAP	GMAP	P@5
TREC 9	LM	-	0.1814	0.0798	0.2839
	RLM	-	0.1853	0.0571	0.2840
	1d KDE	no	0.1983*†	0.0833*†	0.2760
	1d KDE	yes	0.1995*†	0.0848*†	0.3000
	2d KDE	no	0.2042*†	0.0842*†	0.3040
	2d KDE	yes	**0.2046***†	0.0844*†	**0.3120***†
TREC 10	LM	-	0.1625	0.0901	0.3224
	RLM	-	0.1766*	0.0835	0.3592
	1d KDE	no	0.1761*	0.0908	0.3932*†
	1d KDE	yes	0.1792*	0.0934	**0.4000***†
	2d KDE	no	0.1908*†	0.0956	0.3825
	2d KDE	yes	**0.1931***†‡	**0.0992**	0.3959*†

Table: Comparisons between KDE feedback methods (without QE) on

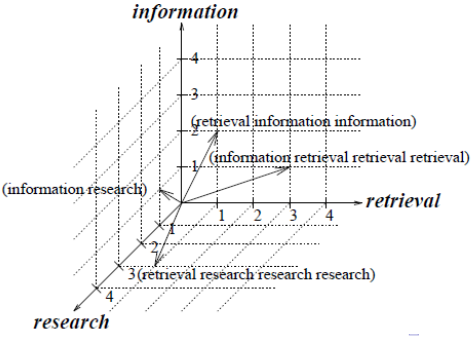
Results with Query Expansion

Dataset	Method	Parameters		Metrics		
		M	*N*	MAP	Recall	P@5
TREC 6	k-NN	n/a	20	0.2175	0.4461	0.3520
	RLM	20	70	0.2634‡	0.5368	0.4360
	1d KDE	10	80	0.2519	0.5311	0.4520
	2d KDE	10	80	**0.2668**†	**0.5420**†‡	**0.4640**†‡
TREC 7	k-NN	n/a	20	0.1614	0.4816	0.3680
	RLM	20	70	0.2151	0.5432	0.4160
	1d KDE	10	80	0.2351†	0.6001†	**0.4425**†
	2d KDE	10	80	**0.2380**	**0.6108**†‡	0.4400
TREC 8	k-NN	n/a	20	0.2320	0.6174	0.4520
	RLM	20	70	0.2701	0.6410	0.4760
	1d KDE	10	80	0.2746	0.6749†	0.4888
	2d KDE	10	80	**0.2957**†‡	**0.6887**†	**0.5120**†‡
TREC Rb	k-NN	n/a	20	0.2575	0.6265	0.4505
	RLM	20	70	0.3304‡	0.8559	0.4949
	1d KDE	10	80	0.3228	0.8725	0.4929
	2d KDE	10	80	**0.3456**†‡	**0.8772**†‡	**0.5152**†‡
TREC 9	k-NN	n/a	10	0.1794	0.6623	0.2512
	RLM	20	70	0.1930	0.6755	0.3233
	1d KDE	10	80	0.1984	0.6851	0.3360
	2d KDE	10	80	**0.2145**†‡	**0.6878**	**0.3562**†‡
TREC 10	k-NN	n/a	10	0.1681	0.7284	0.3123
	RLM	20	70	0.1759	0.7386	0.3347
	1d KDE	10	80	0.2192†	0.7499	0.4004†
	2d KDE	10	80	**0.2213**†	**0.7502**	**0.4204**†

Table: Results of KDE feedback methods with QE. Parameters: *M* (#fdbk docs) and *N* (#expansion terms).

Documents as term vectors

- Terms as dimensions of a document vector (forms an inner product space).
- Inner product $d.q$ gives the similarity between document and query.



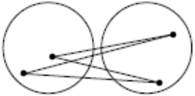
Documents as sets of word embedded vectors

- Each document: A set of real-valued vectors in p dimensions, $D = \{x_i\}_{i=1}^{|D|}, x_i \in \mathbb{R}^p$.
- Need: Generalized distance (inverse similarity) measures, $d(X, Y)$, where X, Y are sets of vectors, which satisfy $d(X, X) = 0, d(X, Y) = d(Y, X)$ and $d(X, Y) + d(Y, Z) \geq d(X, Z)$.

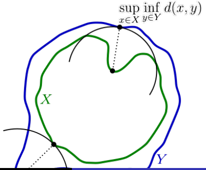
Documents as sets of word embedded vectors

Two distance metrics investigated:

- Average inter-distance:
  $d(X, Y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} d(x, y)$, where $d(x, y)$ is L2 or Euclidean distance between vectors x and y.



- Hausdorff Distance: $d(X, Y) = \max\left(\max_{x \in X} \min_{y \in Y} d(x, y), \max_{y \in Y} \min_{x \in X} d(x, y)\right)$



Illustrative Examples

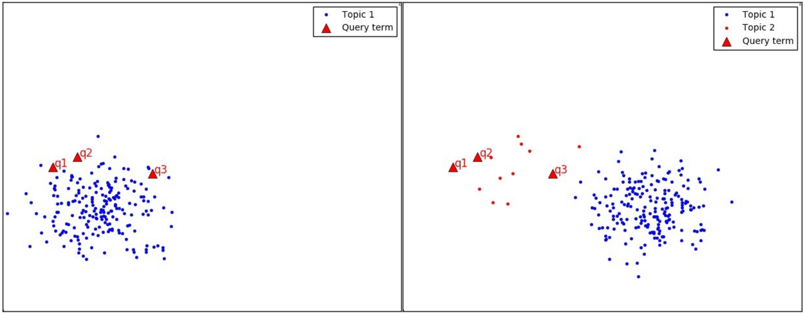


Figure: Two example scenarios of single-topical documents, where the document on the left has a higher similarity to the query than the document on the right.

Illustrative Examples

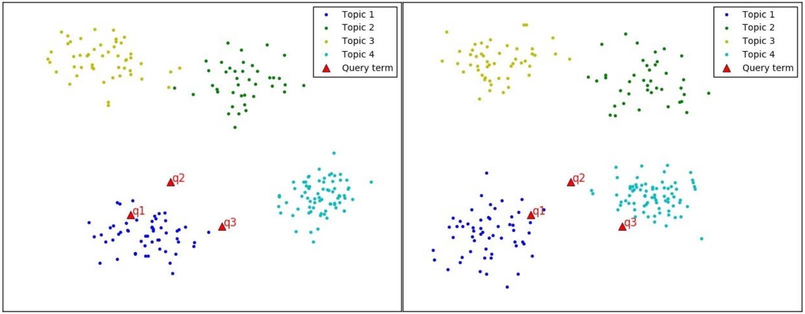


Figure: Two example scenarios where documents are multi-topical, i.e. K-means clustering shows 4 distinct clusters. Document on the right is more similar to the query.

Method Details

- A document is treated as a mixture model of Gaussians of the observed constituent words.
- A query is treated as the observed points drawn from the underlying mixture distribution of a document.
- The query likelihood is then given by the probability of sampling the observed query points from the mixture distribution.

$$sim(q, d) = \frac{1}{K|q|} \sum_i \sum_k q_i \cdot \mu_k \qquad (1)$$

- This is combined with the text based query likelihood (language model based) to obtain the final query likelihood.

$$P(d|q) = \alpha P_{LM}(d|q) + (1-\alpha) P_{WVEC}(d|q) \qquad (2)$$

Practical Considerations for Implementation

- Individually estimating the Gaussian mixture model for each document is time consuming, and slows the indexing process.
- Solution: Cluster the entire vocabulary with an EM based clustering algorithm such as K-means.
- Each term is thus mapped to a cluster id.
- Induce the per-document clusters by grouping together words in a document with the same cluster id and find the centre of each group C_k.

$$\mu_k = \frac{1}{|C_k|} \sum_{x \in C_k} x, C_k = \{x_i : c(w_i) = k\}, i = 1, \ldots, |d| \quad (3)$$

Results

Dataset	Method	Parameters			Metrics			
		Clustered	#clusters	α	MAP	GMAP	Recall	P@5
TREC-6	LM	n/a	n/a	n/a	0.2303	0.0875	0.5011	0.3920
	LM+wvecsim$_{\text{one_cluster}}$	yes	1	0.4	**0.2355**	**0.0918**	**0.5058**	0.3920
	LM+wvecsim$_{\text{no_cluster}}$	no	n/a	0.4	0.2259	0.0827	0.5000	0.3600
	LM+wvecsim$_{\text{kmeans}}$	yes	100	0.4	0.2345	0.0906	0.5027	**0.4040**
TREC-7	LM	n/a	n/a	n/a	0.1750	0.0828	0.4803	**0.4080**
	LM+wvecsim$_{\text{one_cluster}}$	yes	1	0.4	**0.1773**	0.0851	0.4897	0.3960
	LM+wvecsim$_{\text{no_cluster}}$	no	n/a	0.4	0.1664	0.0803	0.4863	0.3640
	LM+wvecsim$_{\text{kmeans}}$	yes	100	0.4	0.1756	**0.0874**	**0.4916**	0.3840
TREC-8	LM	n/a	n/a	n/a	0.2466	0.1318	0.5835	0.4320
	LM+wvecsim$_{\text{one_cluster}}$	yes	1	0.4	0.2541†	0.1465	0.6017	0.4440
	LM+wvecsim$_{\text{no_cluster}}$	no	n/a	0.4	0.2473	0.1396	0.5994	0.4520
	LM+wvecsim$_{\text{kmeans}}$	yes	100	0.4	**0.2558**†	**0.1468**	**0.6017**	**0.4720**
Robust	LM	n/a	n/a	n/a	0.2651	0.1710	0.7803	0.4424
	LM+wvecsim$_{\text{one_cluster}}$	yes	1	0.4	0.2690	0.1701	0.7905	0.4465
	LM+wvecsim$_{\text{no_cluster}}$	no	n/a	0.4	0.2642	0.1646	0.7900	0.4485
	LM+wvecsim$_{\text{kmeans}}$	yes	100	0.4	**0.2804**†	**0.1819**	**0.8010**	**0.4687**

Table: Results of set-based word vector similarities with different settings. K: #clusters, α: weight of the text based query likelihood.

Observations

- Results with word vector based similarities outperform pure text based ones.
- $K = 100$ produces best results for the TREC 8 and the TREC Robust topic sets.
- Show consistent improvements in both recall and precision at top ranks.
- Very fine-grained representation of documents (each constituent word as its own cluster) is not optimal.
- Somewhat surprisingly, $K = 1$, i.e., each document represented by a single point (the average of all words) produces close results to $K = 100$.

Embedded Vector based Multi-modal IR

- Use joint embeddings of text and other data type (e.g. images) to automatically augment text documents with semantically related 'vectors'.
- Example: For a given text document, enhance its representative content (for the purpose of more effective search) by augmenting relevant images from the Wikimedia (Wikipedia image collection).

Embedded Vector based Cross-modal and Cross-lingual IR

- Joint embeddings of vectors can be used for cross-lingual search.
- Individual word embeddings for different languages can be aligned with a parallel corpora.
- Document-Query similarity can be measured on these embedded vector space.
- Joint embeddings can also be used for addressing cross-modal information access, e.g. searching for text documents with image query, searching for speech/video with text query and so on.