中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

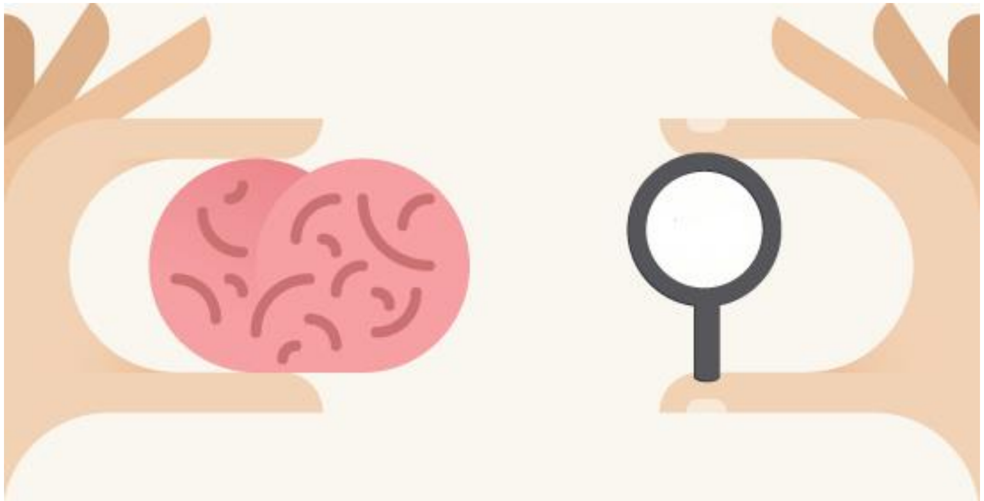
中国科学院网络数据科学与技术重点实验室
Key Laboratory of Network Data Science & Technology ,CAS

Neural Models for Information Retrieval
Part I

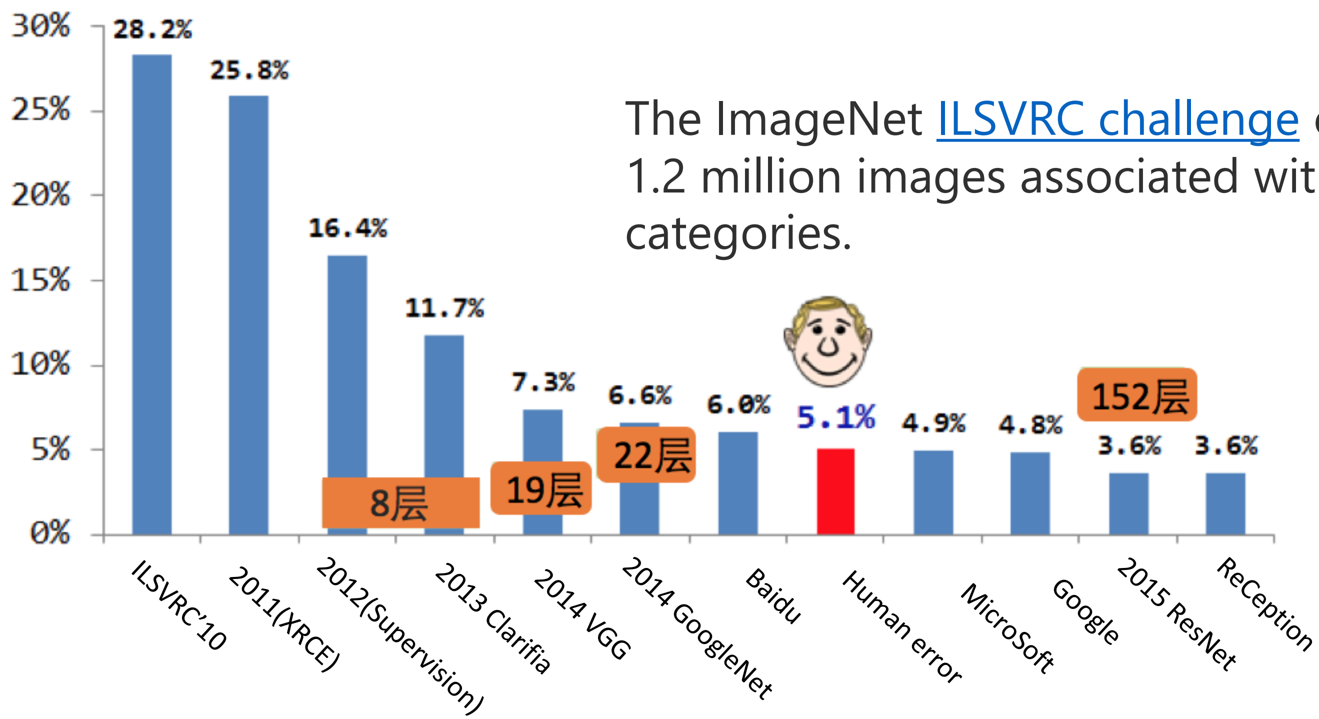
Jiafeng Guo (郭嘉丰), Researcher

Institute of Computing Technology, Chinese Academy of Sciences

Homepage: www.bigdatalab.ac.cn/~gjf

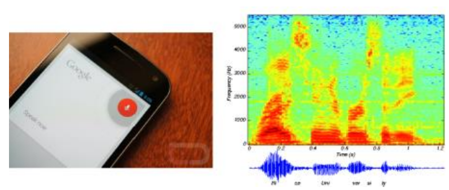


Success stories of deep neural models

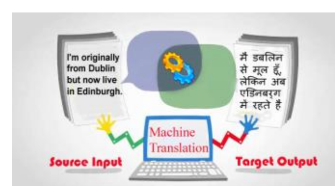


The ImageNet ILSVRC challenge consists of 1.2 million images associated with 1'000 categories.

Object Recognition



Speech Recognition

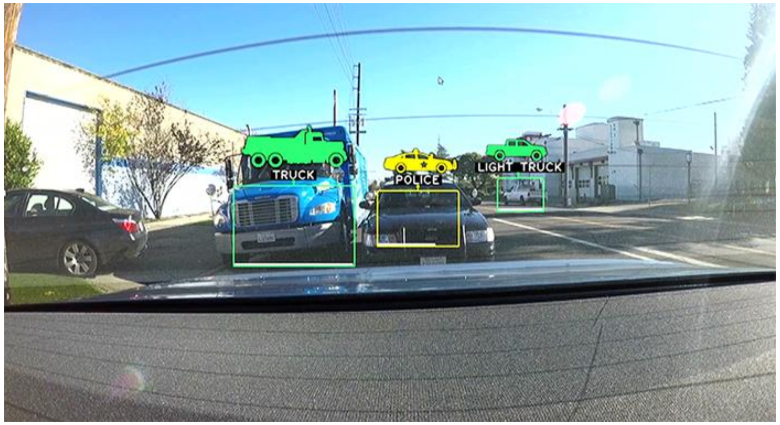


Machine Translation

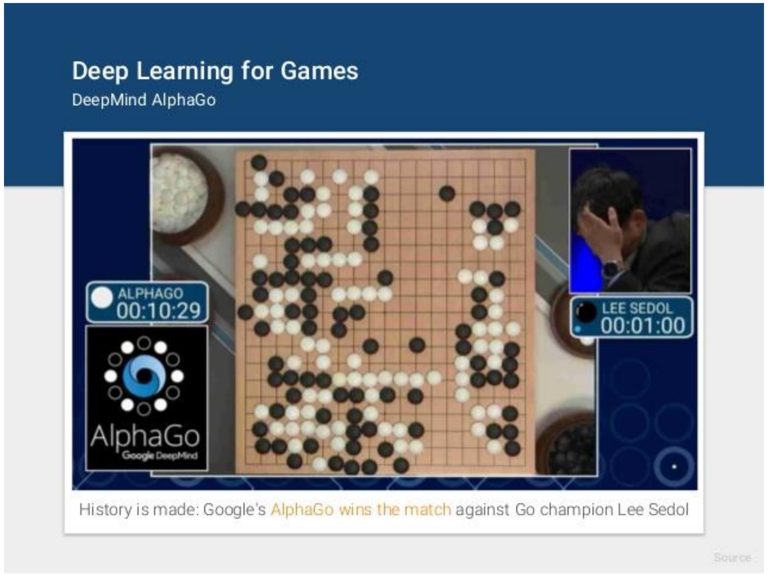
Image Captioning

Success stories of deep neural models

Painting



Driving

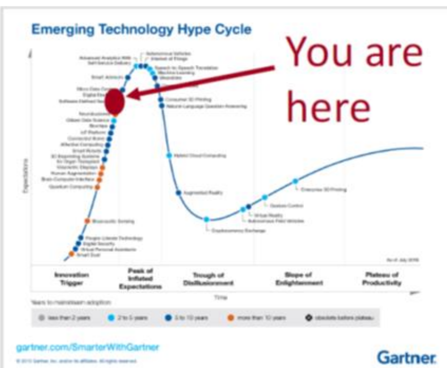


Playing games

Deep Learning for IR

Dominating multiple fields:

2011	2013	2015	2017
speech	vision	NLP	IR



Christopher Manning. [Understanding Human Language: Can NLP and Deep Learning Help?] Keynote SIGIR 2016

SIGIR papers with title words: Neural, Embedding, Convolution, Recurrent, LSTM

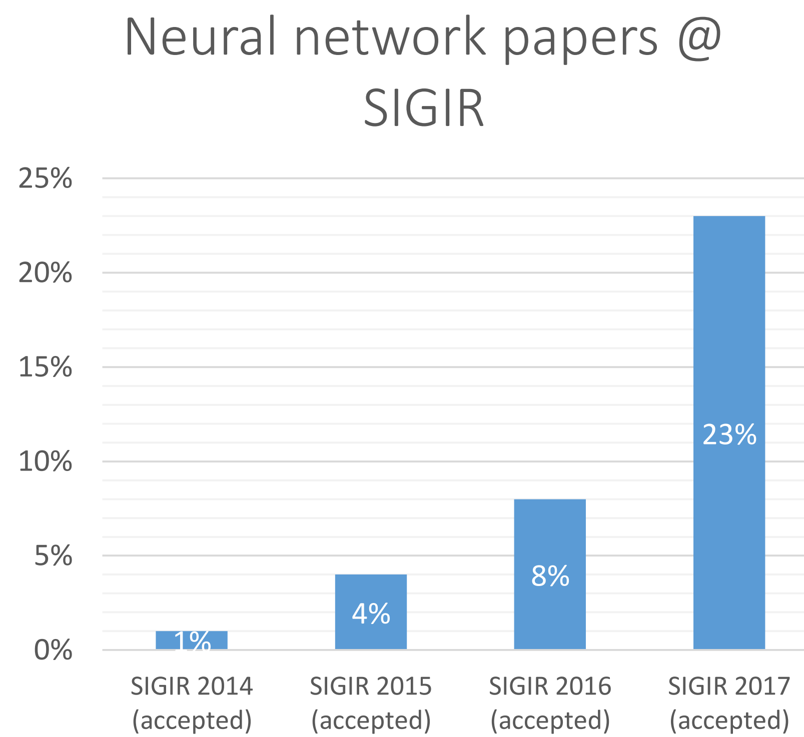


Figure from Mitra & Craswell Tutorial @WSDM 2017

Neural Models for IR

This tutorial mainly focuses on:

- Retrieval of short/long texts, given a text query
- Representation learning
- Shallow and deep neural networks

This presentation includes content from WSDM 2017 tutorial "Neural Text Embeddings for Information Retrieval" by Mitra and Craswell

For broader topics (multimedia, knowledge) see:
Craswell, Croft, Guo, Mitra, and de Rijke. Neu-IR: Workshop on Neural Information Retrieval. SIGIR 2016/SIGIR 2017 workshop

Today's Agenda

Part I

- Fundamentals of IR
- Word Representations
- Word Representations for IR

Part II

- Supervised learning for rank
- Deep neural nets
- Deep neural nets for IR

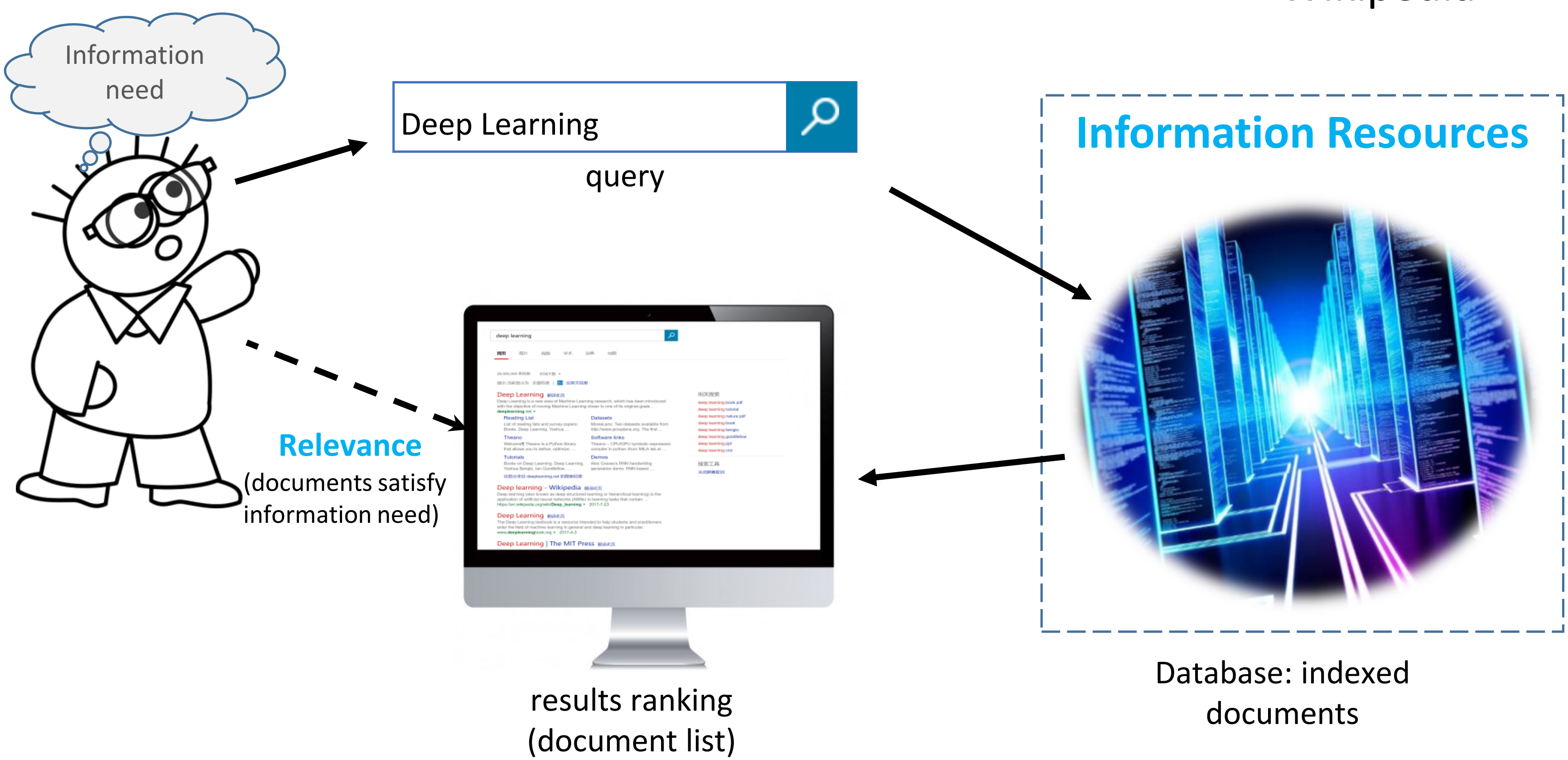
Chapter 1

Fundamentals of IR

Information retrieval (IR) terminology

Information retrieval (IR) is the activity of obtaining information resources **relevant** to an information need from a collection of information resources.

-- Wikipedia



IR Applications

	Ad-hoc retrieval	Question Answering
Query	Keywords	Natural language question
Document	Web page, news article	Supporting passage, entities, facts
TREC experiments	TREC ad hoc	TREC question answering
Evaluation metric	Average precision, NDCG	Mean reciprocal rank
Research solution	Modern TREC rankers BM25, query expansion, learning to rank, links, clicks	IBM@TREC-QA Answer type detection, passage retrieval, relation retrieval, answer processing and ranking
In products	Web search systems: Google, Bing, Baidu, Yandex, …	Watson@Jeopardy
This tutorial	Long text ranking	Short text ranking

Other applications:

- CQA: Similar/related question retrieval
- Conversation: Retrieval response given a sentence

History of IR

- 1950-1960: early days and first empirical observations
  - Hypothesis on automated indexing (Luhn)
  - First experiments and development of guidelines for information retrieval systems evaluation (Cleverdon's Cranfield 1 and Cranfield 2)
  - Early experiments of a Vector Space Model for ranking (Salton's SMART)
- 1970-1980: active development of information retrieval
  - Establishment of a Vector Space Model for ranking
  - Ranking models based on probability ranking principles (PRP)
- 1990s: further development and formalization of IR (new applications and theoretical explanations)
  - Statistical Language Models (Croft' 98)
  - Development of large scale collections for IR system evaluation (TREC)
- 2000s: web search, large scale search engine in the wild, anti-spam
  - Machine Learning to Rank
  - MapReduce, GFS, Hadoop …
- 2010s: entity search, social search, real-time search

Challenges in (neural) IR [1/4]

- Vocabulary mismatch

  Q: How many people live in Sydney?

  ➢ Sydney's population is 4.9 million
  [relevant, but missing 'people' and 'live']

  ➢ Hundreds of people queueing for live music in Sydney
  [irrelevant, and matching 'people' and 'live']

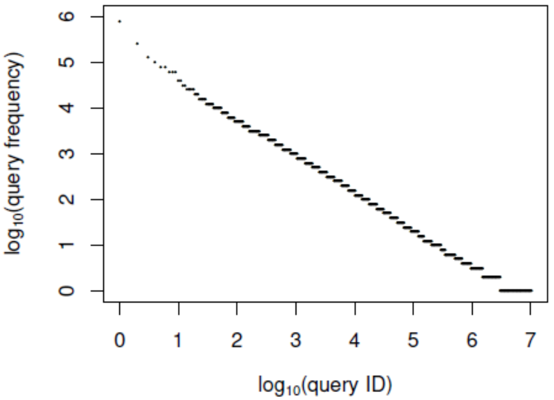
Vocab mismatch:

- Worse for short texts
- Still an issue for long texts

- Robustness to rare inputs

  ➢ More than 70% of the distinct query are seen only once

  ➢ Q: "pekarovic land company"



Learning good representation of text is important for dealing with vocabulary mismatch, but exact matching is also important to deal with rare terms and intents.

Challenges in (neural) IR [2/4]

- Q and D vary in length
  - ➢ Models must handle short (keyword) queries and long (verbose) queries
  - ➢ Models must handle varied length documents

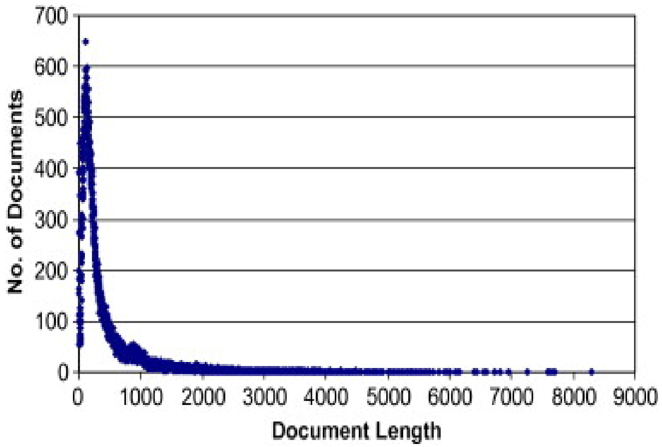


Figure from: AleAhmad, Abolfazl, et al. "Hamshahri: A standard Persian text collection." *Knowledge-Based Systems* 22.5 (2009): 382-387.

- Different hypothesis about long document [Roberson et al. 1994]
  - ❑ **Verbosity hypothesis** : Long document covering a similar scope but with more words.
  - ❑ **Scope hypothesis** : long document consists of a number of unrelated short documents concatenated together.

A good retrieval model should be able to handle and robust to varied length queries and documents

Challenges in (neural) IR [slide 3/4]

- Need to learn Q-D relationship that generalizes to the tail
  - ➢ Unseen Q
  - ➢ Unseen D
  - ➢ Unseen information needs
  - ➢ Unseen vocabulary
- Robustness to corpus variance
  - ➢ Simple model vs. deep models
  - ➢ “Out of box” performance
  - ➢ Overfitting

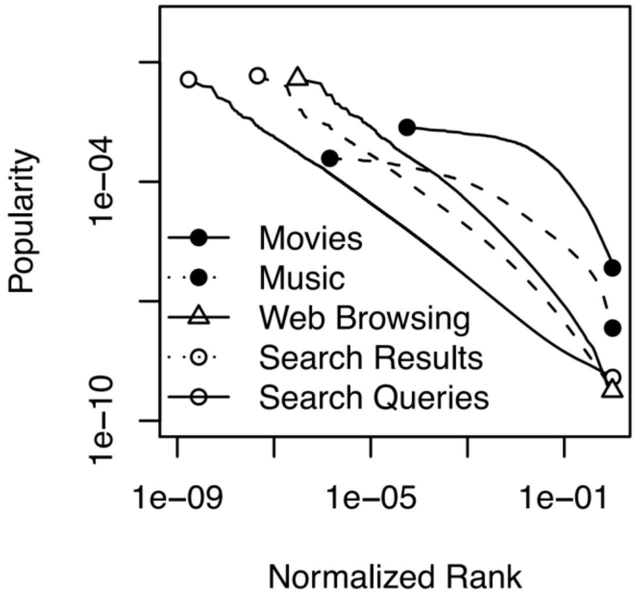


Figure from: Goel, Broder, Gabrilovich, and Pang. [Anatomy of the long tail: ordinary people with extraordinary tastes](). WWW Conference 2010

A good retrieval model should be able to capture the essential relevance patterns between query and document, and generalize well on unseen data

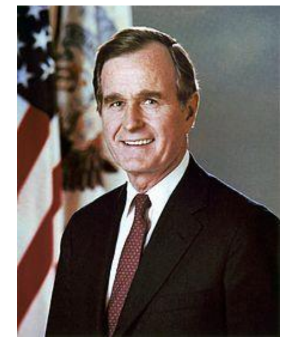
Challenges in (neural) IR [4/4]

- Need to interpret words based on context (e.g., temporal)

query:
"United States president"

Today

Recent



In older (1990s) TREC data

- Robustness to errors in input
  - ➢ Traditional IR models: specific components for error corrections
  - ➢ Neural IR models: character-level operation and/or representation learning from noisy data
- Efficient retrieval over many documents
  - ➢ Inverted files, KD-Tree, LSH, …

Popular IR Metrics

IR metrics focus on rank-based comparison of the retrieved result set R to an ideal ranking of documents, as determined by manual judgments or implicit feedback from user behavior data.

1. Precision and recall

$$Precision_q = \frac{\sum_{\langle i,d\rangle \in R_q} rel_q(d)}{|R_q|} \qquad Recall_q = \frac{\sum_{\langle i,d\rangle \in R_q} rel_q(d)}{\sum_{d \in D} rel_q(d)}$$

2. Mean reciprocal rank (MRR)

$$RR_q = \max_{\langle i,d\rangle \in R_q} \frac{rel_q(d)}{i}$$

3. Mean average precision (MAP)

$$AveP_q = \frac{\sum_{\langle i,d\rangle \in R_q} Precision_{q,i} \times rel_q(d)}{\sum_{d \in D} rel_q(d)}$$

4. Normalized discount cumulative gain (NDCG)

$$\mathrm{DCG}_q = \sum_{\langle i,d\rangle \in R_q} \frac{2^{rel_q(d)} - 1}{\log_2(i+1)} \qquad \mathrm{N}DCG_q = \frac{DCG_q}{IDCG_q}$$

Traditional IR Models

1. Boolean models (Lancaster et al., 1973):
   - Simple model based on set theory
   - Queries specified as boolean expressions
2. Vector Space models (Salton et al., 1983):
   - Unique terms that form the VOCABULARY
   - These “orthogonal” terms form a vector space.
3. Probabilistic models:
   - BM25 (Robertson et al., 1994)
   - Language model (Croft et al., 1998)
   - Translation models (Berger and Lafferty, 1999)
   - Dependence model (Metzler and Croft, 2005)
4. Pseudo relevance feedback (Lavrenko, 2008, Lavrenko and Croft, 2001)
   - Execute an additional round of retrieval

Learning to Rank

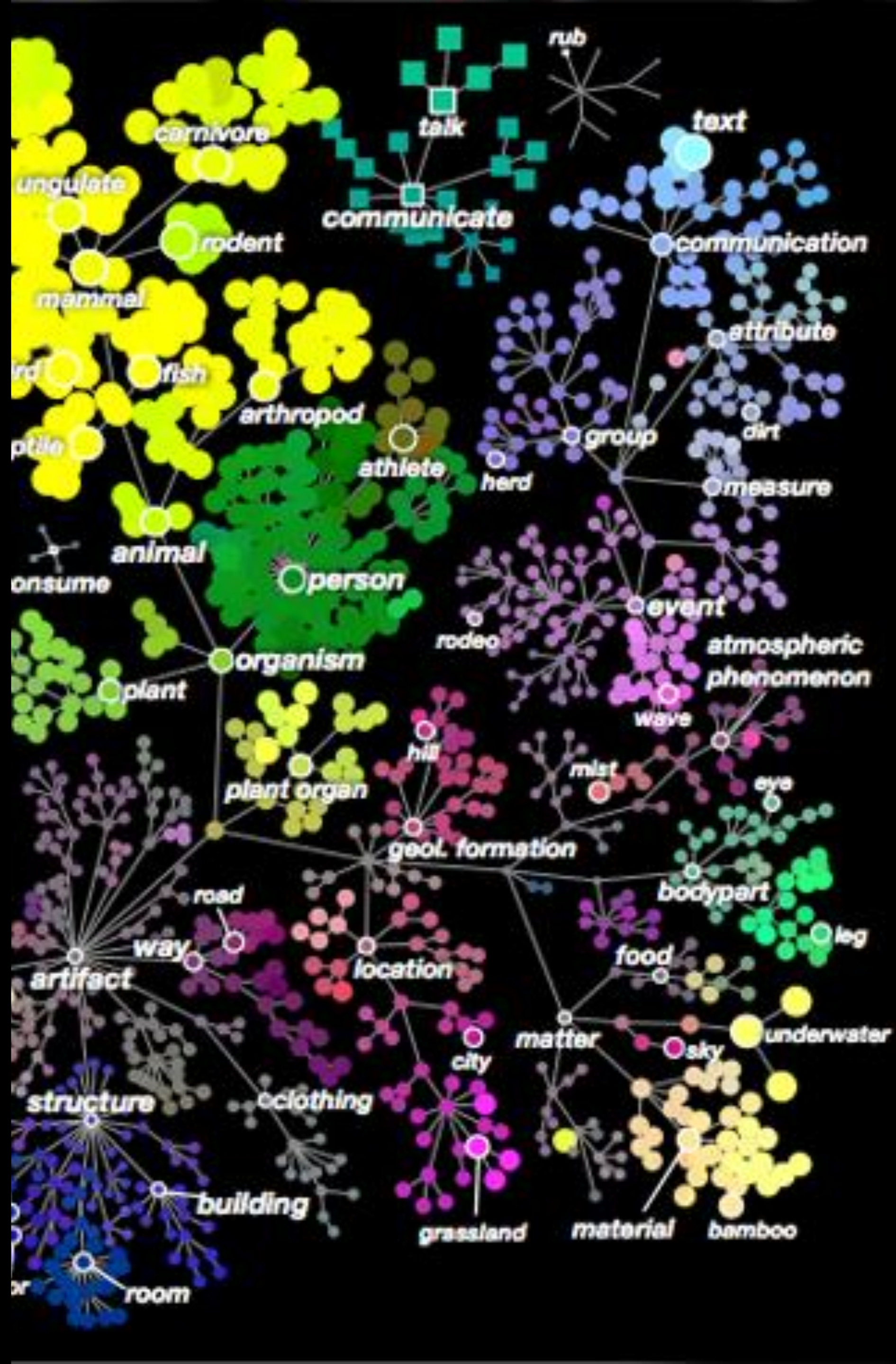
	1995	1998	2001	2002	2003	2005	2006	2007	2008	2009	2010	2011	2012	2017
Univ. of California	LTIR													
New York Univ.							P-norm push							
Hebrew Univ.			Pranking		RankBoost									
Cornell Univ.				RankSVM				SVMMAP						
Columbia Univ.					RankBoost									
Univ. of Toronto										BoltzRank				
Northeastern Univ.										Doc Selection				
Princeton Univ.					RankBoost									
Nottingham Univ.														ES-Rank
ICT, CAS													Top-K	
AT&T		LTOT												
Yahoo!						SubsetRanking		GBRank			SmoothRank	LTRC		
Microsoft						RankNet	LambdaRank	McRank, Frank, Adarank, ListNet, LETOR	GlobalRanking, SoftRank, ListMLE	Consistency, Generalization, Judgment	Lambda MART	LambdaGradient		
Google									RTC					
Yandex												TR Learning		

Neural Approaches to IR

- Related document search:
  - semantic hashing, Salakhutdinov and Hinton (2009)…
- Ad-hoc search:
  - **Word Representation based models**
    - FV, Clinchant, S. and Perronnin, F. (2013),
    - QLM, Sordoni et al. (2014);
    - NWT, Guo et al. (2016)
  - **Neural network based models**
    - Title/Snippet-based search, DSSM, Huang et al. (2013); …
    - Different Granularity search: Cohen et al. (2016);
    - Full document search: DRMM, Guo et al. (2016).
- Wider adoption in IR from 2015:
  - QA/CQA, query completion, query suggestion, sponsored search

Chapter 2

Word Representations



Local Representation of Words

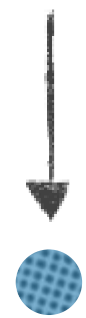
- Words are the building blocks of texts
- Traditional IR often treats words as atomic symbols:

Man | Woman | Dog | Computer

- also known as “one-hot” or local representation

One-Hot Representation	
man	[1,0,...,0,0,...,0,0]
woman	[0,1,...,0,0,...,0,0]
dog	[0,0,...,1,0,...,0,0]
computer	[0,0,...,0,0,...,1,0]

man



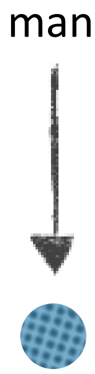
- local representation: each word is locally represented by a distinct node.

Limitations of Local Representations

- Local representation makes a strong independent assumption between words

Local Representation	
man	[1,0,…,0,0,…,0,0]
woman	[0,1,…,0,0,…,0,0]
car	[0,0,…,1,0,…,0,0]
automobile	[0,0,…,0,0,…,1,0]

cos(car, automobile) = 0!



- Local representation is not efficient
  - ➢ require N nodes to represent N words

Limitations of Local Representations

- Local representations are arbitrary, and cannot generalize between words
  - ➢ The model can leverage very little of what it has learned about “groups” when it is processing data about “teams”

Training corpus:

- There are three teams left for the qualification.
- four teams have passed the first round.
- four groups are playing in the field.

Assign a probability to an unseen bigram “three groups”:

$$p(\text{groups} \mid \text{three}) = 0!$$

No generalization

“The first thing you do with a word symbol is you convert it to a word vector. And you learn to do that, you learn for each word how to turn a symbol into a vector, say, 300 components, and after you’ve done learning, you’ll discover the vector for Tuesday is very similar to the vector for Wednesday.”

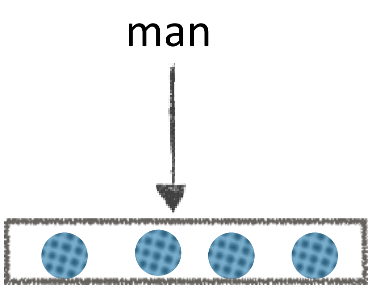
– Geoffrey E. Hinton

Deep Learning.
Royal Society keynote
recorded May 22, 2015

Distributed Representation of Words

- Vector space models (VSM) represent (embed) words in a continuous vector space
- also known as distributed representations[1]: all the words share all the nodes

Vector Space Representation	
man	[0.326172, . . . , 0.00524902, . . . , 0.0209961]
woman	[0.243164, . . . , −0.205078, . . . , −0.0294189]
car	[0.0512695, . . . , −0.306641, . . . , 0.222656]
automobile	[0.107422, . . . , −0.0375977, . . . , −0.0620117]



Vectors from GoogleNews-vectors-negative300.bin

Hinton, G. E., et al. Distributed representations. In Rumelhart, D. E., McClelland, J. L., and PDP Research Group, C., editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1,1986, pages 77–109. MIT Press, Cambridge, MA, USA.

Pros of Distributed Representations

- Distributed representations
  - Semantically similar words are mapped to nearby points

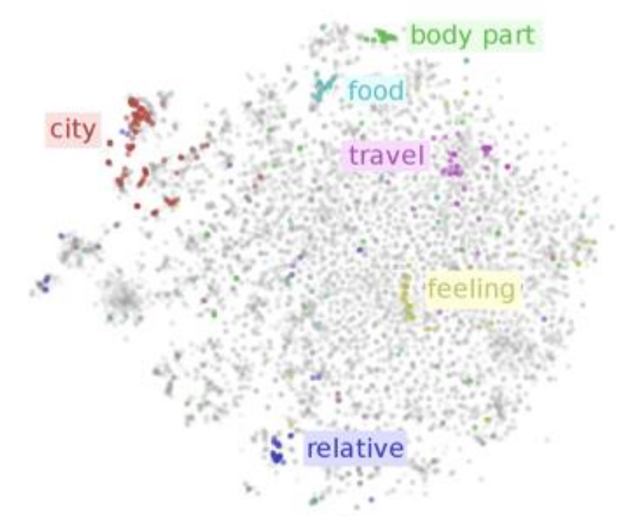
Distributed Representation	
man	[0.326172, . . . , 0.00524902, . . . , 0.0209961]
woman	[0.243164, . . . , −0.205078, . . . , −0.0294189]
car	[0.0512695, . . . , −0.306641, . . . , 0.222656]
automobile	[0.107422, . . . , −0.0375977, . . . , −0.0620117]

cos(man, women) = 0.77

cos(man, automobile) = 0.25

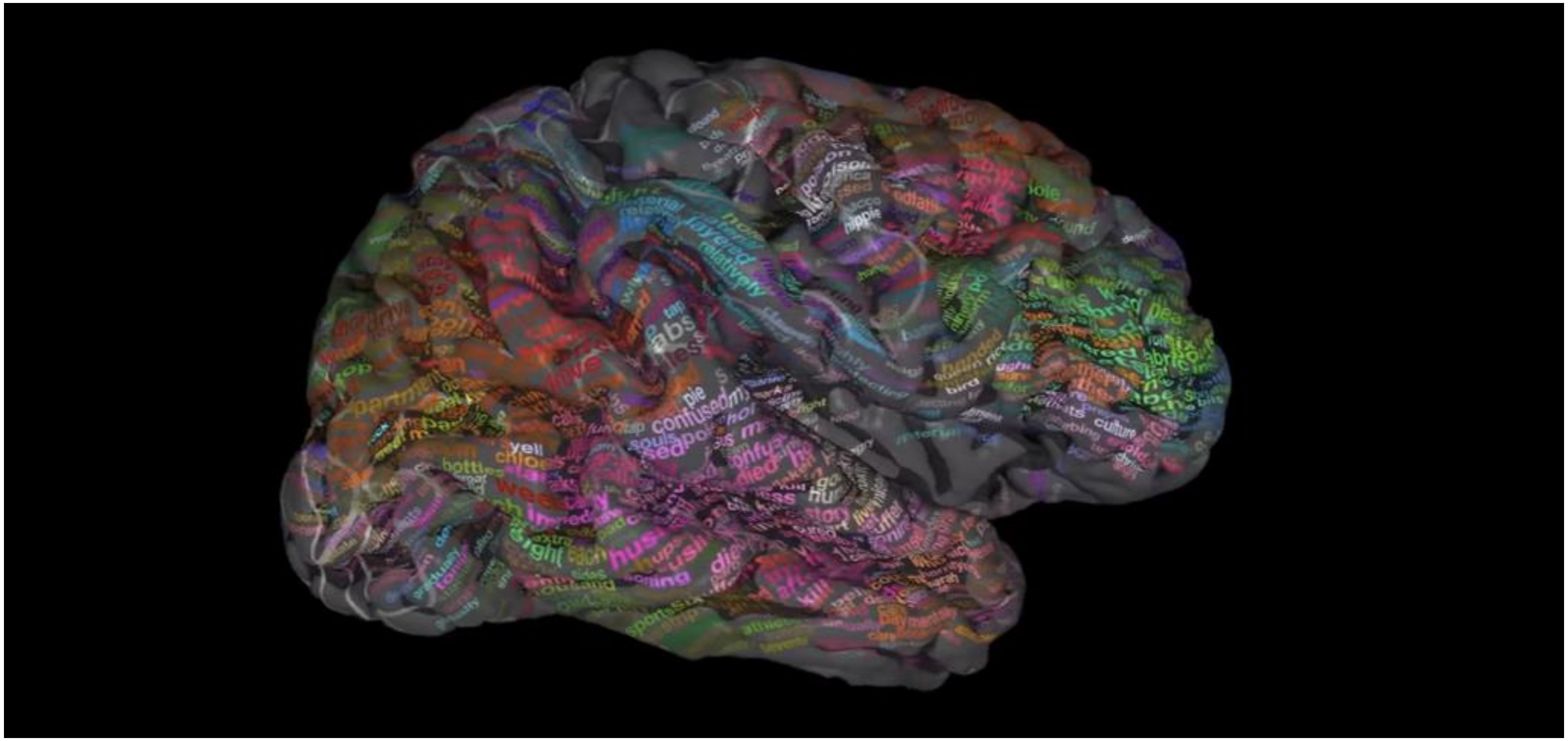
Pros of Distributed Representations

FRANCE 454	JESUS 1973	XBOX 6909	REDDISH 11724	SCRATCHED 29869	MEGABITS 87025
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES



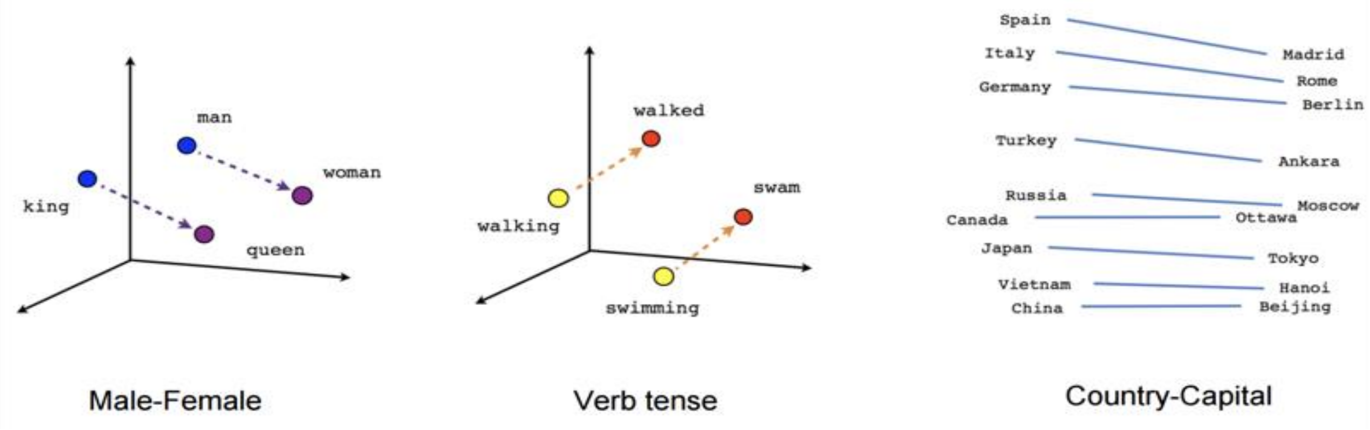
What words have embeddings closest to a given word?
From Collobert et al. (2011)

The Brain Dictionary



Huth, A.G., de Heer, W.A., Griffiths, T.L., Theunissen, F.E., Gallant, J.L., 2016. Natural speech reveals the semantic maps that tile human cerebral cortex. Nature 532, 453–458. doi:10.1038/nature17637

Pros of Distributed Representation

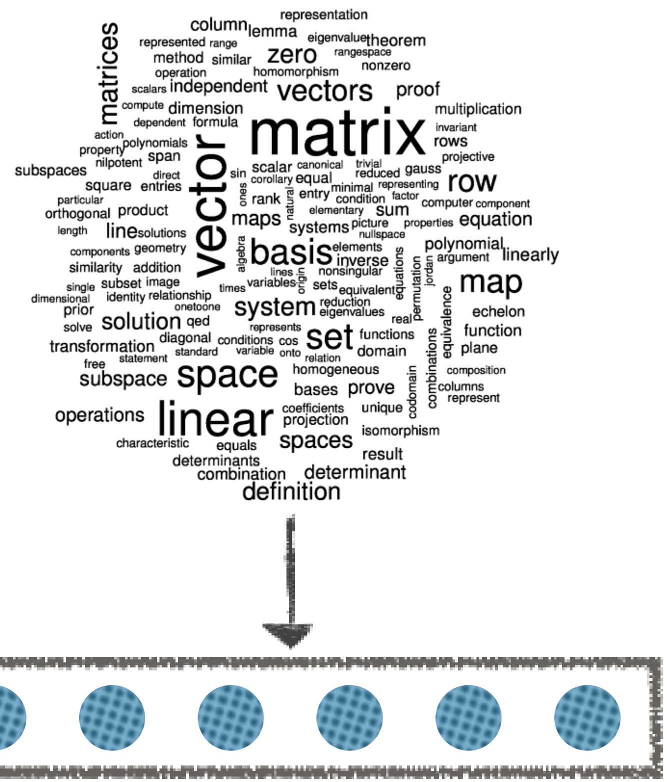


Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

From Tomas Mikolov et al. Efficient estimation of word representations in vector space. In Proceedings of Workshop of ICLR, 2013.

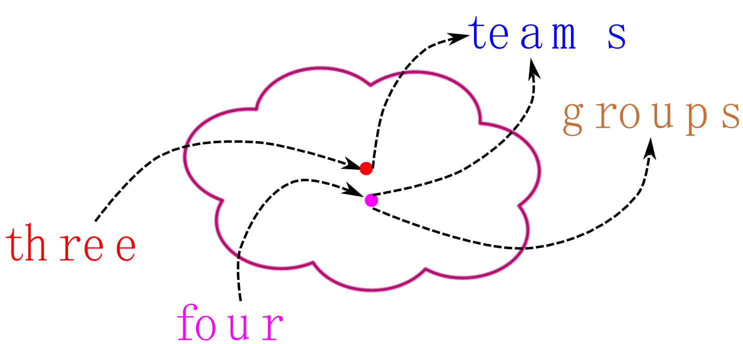
Pros of Distributed Representation

- Distributed representation is efficient
  - N nodes can represent 2^N words (binary case)



Pros of Distributed Representation

- Distributed representations can generalize between words
  - Semantically similar words are mapped to nearby points



Training corpus:

- There are three teams left for the qualification.
- four teams have passed the first round.
- four groups are playing in the field.

Assign a probability to an unseen bigram “three groups”:

p(teams|three) > 0 p(groups|three) > 0

- Generalization ability: language model using distributed word representation can assign a reasonable probability

"The gains so far have not so much been from true Deep Learning (use of a hierarchy of more abstract representations to promote generalization) as from the use of distributed word representations—through the use of real-valued vector representations of words. Having a dense, multi-dimensional representation of similarity between all words is incredibly useful in NLP…"

– Christopher D. Manning

Computational Linguistics and Deep Learning. Computational Linguistics, 41(4):701–707, 2015.

How to learn word representations?

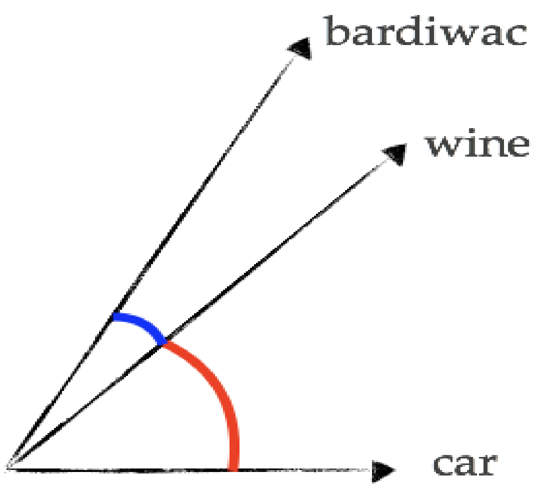
What is the meaning of “bardiwac ”?

the doctor. `Just checking on the	**bardiwac**	,' he boomed as he came back. `Edith's very
`I hope you'll take to a good French	**bardiwac**	,' chimed in Arthur Iverson jovially. `One
`Our host did slip out to attend to the	**bardiwac**	…' `That was before the shrimp
Iverson did when he went through to see to the	**bardiwac**	before dinner.' Henry rubbed his hands.
and drinking red wine from France -- sour	**bardiwac**	, which had proved hard to sell. The room
eyes were alight and he was drinking the	**bardiwac**	down like water. `It is like Hallow-fair
quizzically at him and offering him some more	**bardiwac**	. He shook his head. `I will sleep
drinks (as Queen Victoria reputedly did with	**bardiwac**	and malt whisky), but still the result
Do we really `wash down' a good meal with	**bardiwac**	? Port is immediately suggested by Stilton
completely different: cheap and cheerful	**bardiwac**	. Two good examples from Victoria Wine are
examples from Victoria Wine are its house	**bardiwac**	, juicy and a touch almondy, a good buy
opened a bottle of rather rust-coloured	**bardiwac**	. I ate too much and drank nearly three-quarters
elections, it was apparent the SDP of `	**bardiwac**	and chips' mould-breaking fame at the time
the black hills. Not a night of vintage	**bardiwac**	. Burnley: Pearce, Measham, McGrory
SONS Old School -- the Marlborian navy,	**bardiwac**	and slim-white stripe. Heavy woven silk
white-hot passion. We are like a good bottle of	**bardiwac**	; we both have sediment in our shoes. </p>
few minutes later he was uncorking a fine	**bardiwac**	in Masha's room, saying he had something
the phone. Surkov silently offered me more	**bardiwac**	but I indicated a bottle of Perrier. </p>
defenders as Villa swept past them like a	**bardiwac**	and blue tidal wave. Things are difficult
campaign. Refreshed by a nimble in-flight	**bardiwac**	, they serenaded him with a special song

From Stefan Evert. Distributional Semantic Models . NAACL-HLT 2010 Tutorial

Distributional Semantics in a Nutshell

	glass	drink	grape	red	meal
bardiwac	10	22	43	16	29
wine	14	10	4	15	45
car	5	0	0	10	0



Distributional Hypothesis

" Words that occur in the same contexts tend to have similar meanings."

-- Zellig Harris [Harris, 1954]

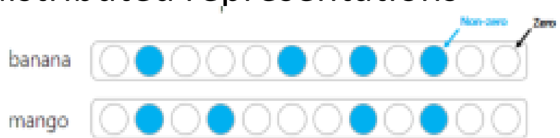
" A word is characterized by the company it keeps."

-- Firth, J. R. [Firth, 1957]

Distributed and distributional

- **Distributed representation**: Vector represents a concept as a pattern, rather than 1-hot

Distributional methods use distributed representations

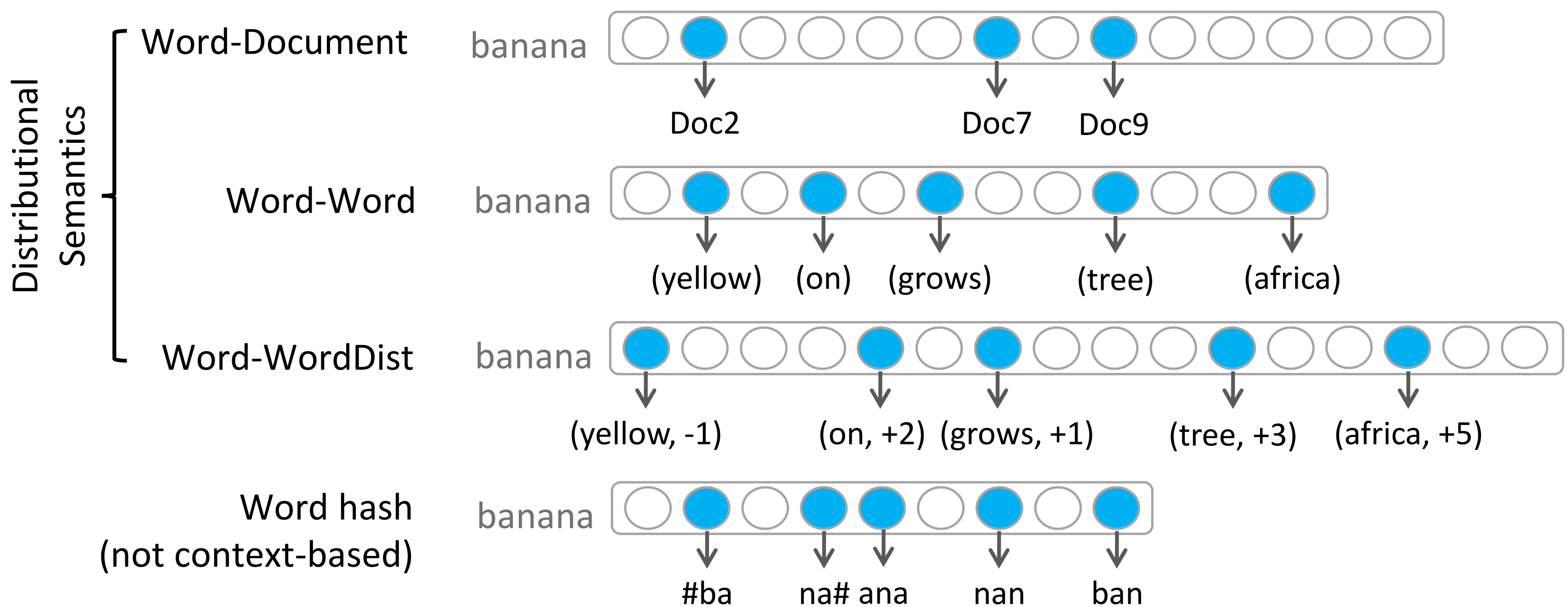


- **Distributional semantics**: Linguistic items with similar distributions (e.g. context words) have similar meanings

"You shall know a word by the company it keeps"

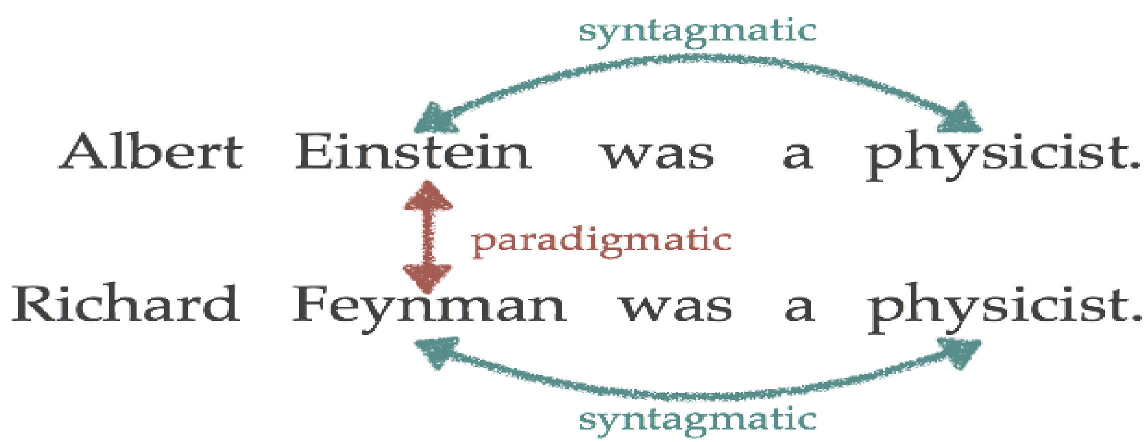
Context is the key

Context is the key in distributional hypothesis.
What type of context you use decides what kind of meaning or semantic relations between words you obtain.



Two tales of semantic relationships

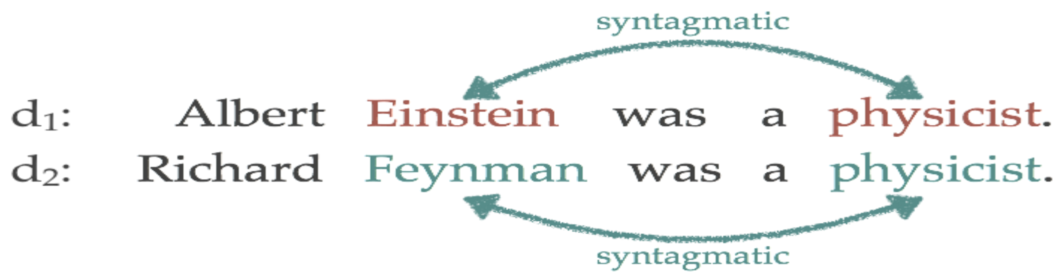
- Syntagmatic (or topical) relations: concerning positioning, and relate words that co-occur in the same text region.
- Paradigmatic (or typical) relations: concerning substitution, and relate words that occur in the same context but not at the same time.



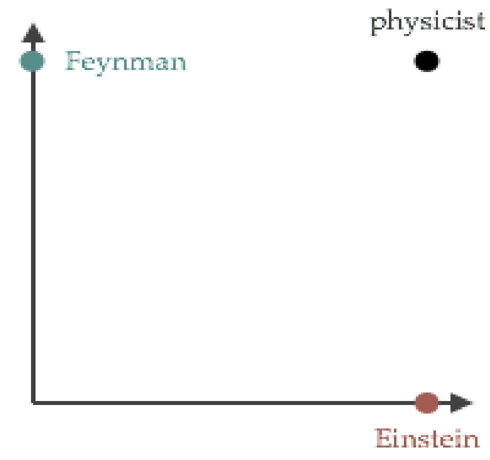
Sahlgren, M. (2008). The distributional hypothesis. Italian Journal of Linguistics, 20(1):33–54.
Fei Sun et al. Learning Word Representations by Jointly Modeling Syntagmatic and Paradigmatic Relations. In Proceedings of ACL. 2015, 136–145

Syntagmatic Models

Distributional models with syntagmatic relations collect information about which context regions words occur

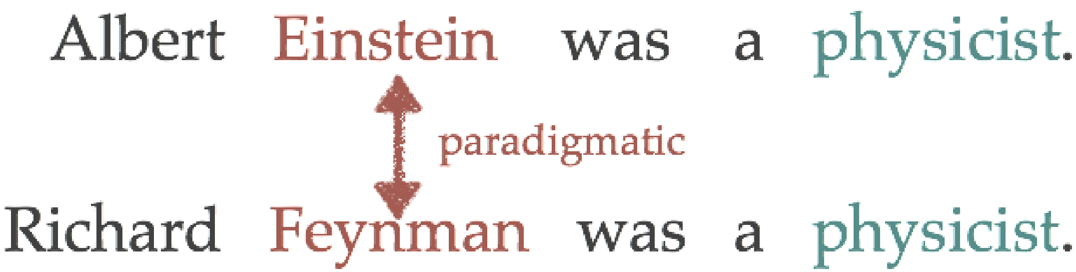


	d_1	d_2
Einstein	1	0
Feynman	0	1
physicist	1	1

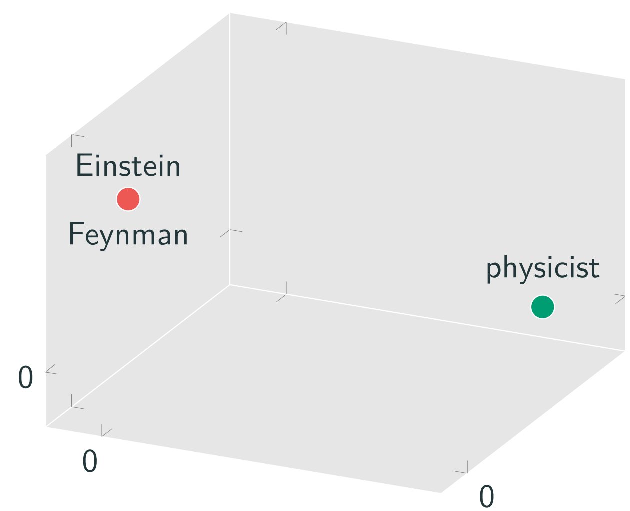


Paradigmatic

Distributional models with paradigmatic relations collect information about which other words surround a word



	Einstein	Feynman	physicist
Einstein	0	0	1
Feynman	0	0	1
physicist	1	1	0



The refined distributional hypothesis: “A distributional model accumulated from co-occurrence information contains syntagmatic relations between words, while a distributional model accumulated from information about shared neighbors contains paradigmatic relations between words.”

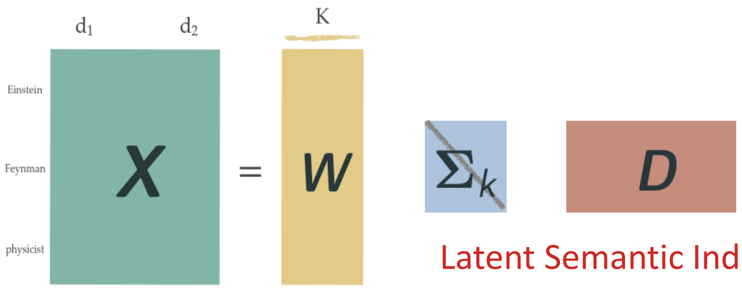
Distributed Representation

- Explicit vector representations
  - Vector space model based on raw counts of context features
  - Highly sparse and high dimensional

- Embedding
  - A representation of items in a new space such that the relationships between the items are preserved from the original representation
  - A simpler representation
    - A reduction in the number of dimensions
    - An increase in the sparseness of the representation
    - Disentangling the principle components of the vector space
    - A combination of these goals

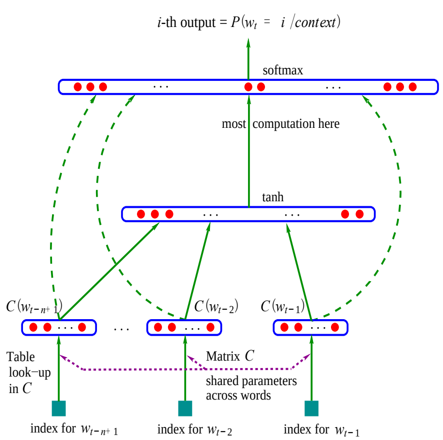
Word Embedding Models

Syntagmatic relations

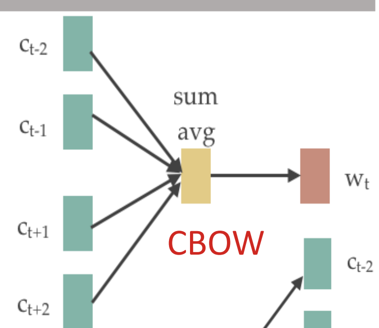


Latent Semantic Indexing

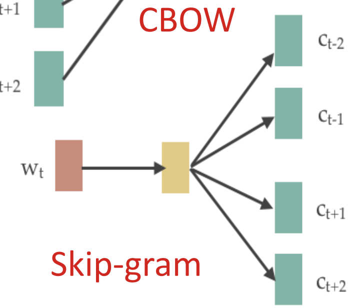
Paradigmatic relations



Neural Probabilistic Language Model

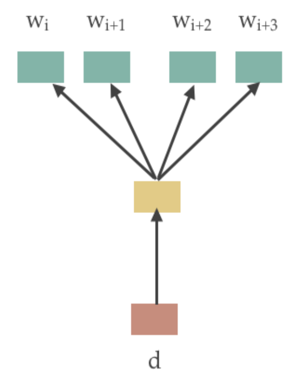


CBOW

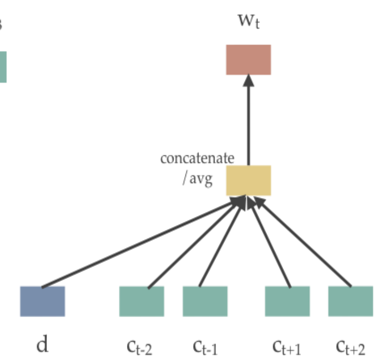


Skip-gram

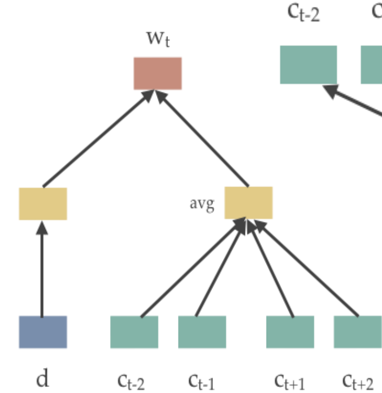
Joint relations



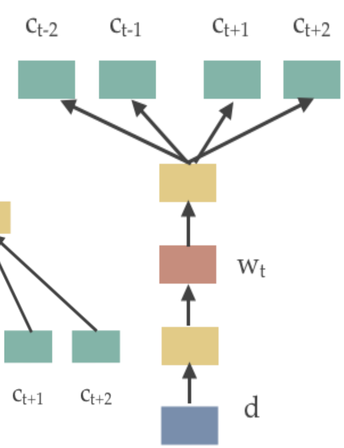
PV-DBOW



PV-DM



HDC



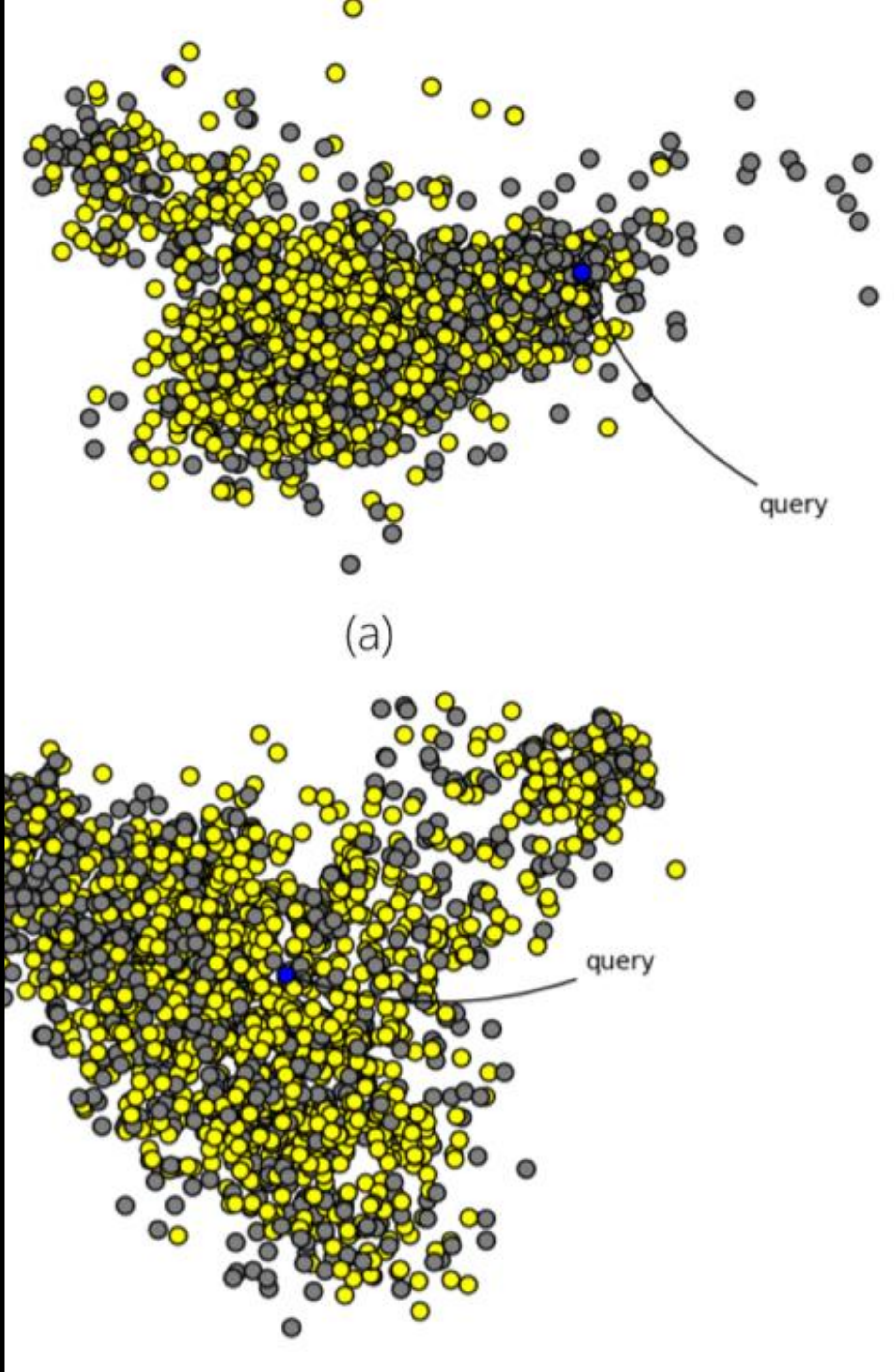
PDC

Discussion of word representations

- Distributed representation is learned based on distributional hypothesis
  - ➢ Weighted counts, matrix factorization, neural embedding
- Choice of contexts affects semantics
  - ➢ Syntagmatic vs Paradigmatic
- Efficiency (i.e. large data) weights more than complex model
  - ➢ Scalability could be the key advantage of neural word embeddings

Chapter 3

Word Embeddings for IR



Traditional IR Foundations

Retrieval based on local representations (BoW)

peace process in the Middle East

As for the Arabian and Palestinean voices that are against the current negotiations and the so-called peace process, they are not against peace per se, but rather for their well-founded predictions that Israel would NOT give an inch of the West bank (and most probably the same for Golan Heights) back to the Arabs. An 18 months of "negotiations" in Madrid, and Washington proved these predictions. Now many will jump on me saying why are you blaming israelis for no-result negotiations. I would say why would the Arabs stall the negotiations, what do they have to loose ?

peace	1
Process	1
Middle	1
East	1

Bag-of-words Representation

Syntactic Matching

Arabian	1
predictions	2
peace	2
negotiations	4
…	…

Bag-of-words Representation

When Word Embedding Comes…

Retrieval based on distributed representations (BoWE)

peace process in the Middle East

One-hot embedding	idf
[0.1 0.3 0.03 0 0.4 …. 0.05 0.12 0.02]	*4.8*
[0.2 0.13 0.03 0 0 …. 0.07 0.09 0.01]	1.2
[0.13 0.3 0.3 0 0.2 …. 0.08 0.87 0.02]	*1.4*
[0.3 0.4 0.09 0 0.3 …. 0.05 0.34 0.14]	1.6

Bag-of-word-embedding Representation

Semantic Matching

As for the Arabian and Palestinean voices that are against the current negotiations and the so-called peace process, they are not against peace per se, but rather for their well-founded predictions that Israel would NOT give an inch of the West bank (and most probably the same for Golan Heights) back to the Arabs. An 18 months of "negotiations" in Madrid, and Washington proved these predictions. Now many will jump on me saying why are you blaming israelis for no-result negotiations. I would say why would the Arabs stall the negotiations, what do they have to loose ?

One-hot embedding	idf
[0.4 0.3 0.45 0 0.3 …. 0.09 0.24 0.7 0.01]	*1.2*
[0.2 0.1 0.03 0 0.1 …. 0.18 0.91 0.2 0.02]	1.3
[0.1 0.3 0.13 0 0.1 …. 0.16 0.25 0.8 0.03]	*2.5*
[0.3 0.5 0.11 0 0.2 …. 0.03 0.170.1 0.15]	5.2
[0.7 0.9 0.01 0 0.6 …. 0.15 0.35 0.4 0.26]	*1.3*
…	…

Bag-of-word-embedding Representation

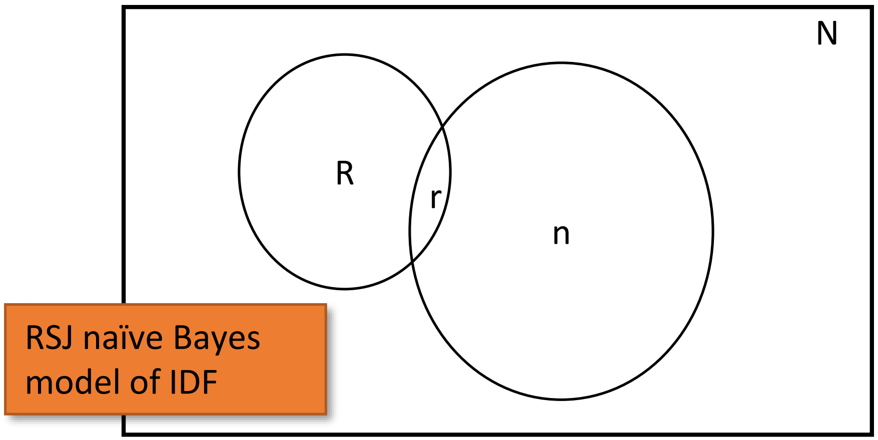
How to incorporate embeddings

1. Extend traditional IR models
   - ➢ Term weighting, language model smoothing, translation of vocab
2. IR models that work in the embedding space
   - ➢ Centroid distance, word mover’s distance
3. Expand query using embeddings (followed by non-neural IR)
   - ➢ Add words similar to the query

Traditional Term Weighting

- **Inverse document frequency**

Robertson and Sparck-Jones (1977)

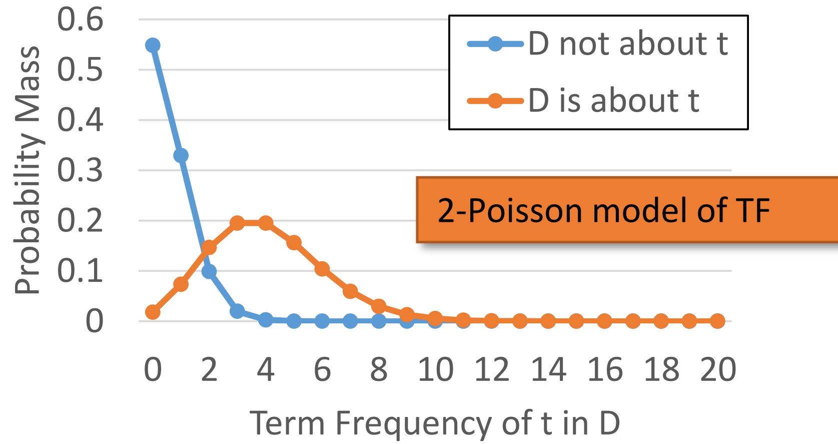


$$w_i^{\mathrm{RSJ}} = \log \frac{(r_i + 0.5)(N - R - n_i + r_i + 0.5)}{(n_i - r_i + 0.5)(R - r_i + 0.5)}$$

$$w_i^{\mathrm{IDF}} = \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

- **Term frequency**

Harter (1975)

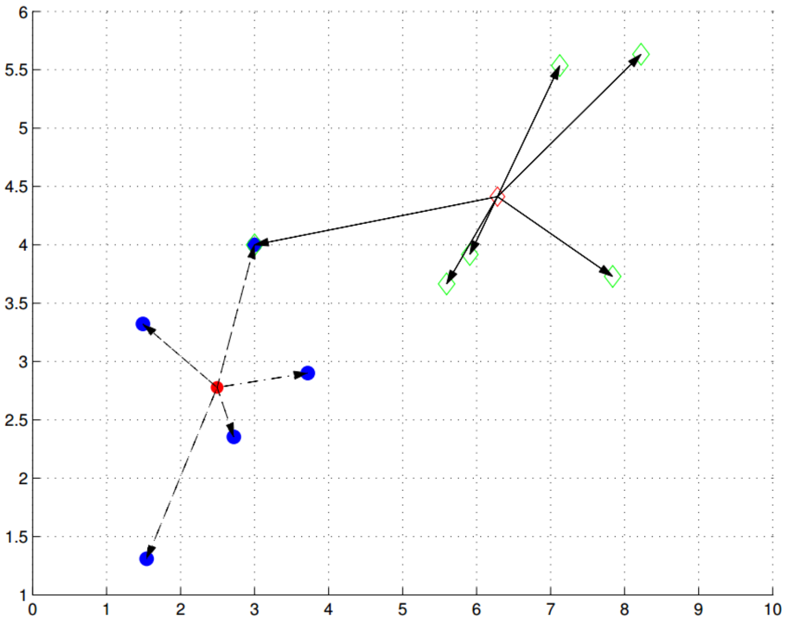


- Adjust TF model for **doc length**

Rank D by: $\sum_{t \in Q} TF(t, D) * IDF(t)$

Robertson and Zaragoza. [The probabilistic relevance framework: BM25 and beyond](). *Foundations and Trends® in Information Retrieval* 3, no. 4 (2009)

Term weighting using word embeddings



$y = \frac{r}{R}$ (term recall)

- Fraction of positive docs with t
- The r and R were missing in RSJ

$\boldsymbol{x} = \boldsymbol{t} - \overline{\boldsymbol{q}}$

- $\boldsymbol{t}$ is embedding of t
- $\overline{\boldsymbol{q}}$ is centroid of all query terms

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1$$

- Weight TF-IDF using $\hat{y}$

Zheng and Callan. Learning to reweight terms with distributed representations. SIGIR 2015

Term weighting using word embeddings

Performance with language model:

Query Model	ROBUST04 P@10	ROBUST04 MAP	WT10g P@10	WT10g MAP	GOV2 P@10	GOV2 MAP	ClueWeb09B P@10	ClueWeb09B MAP
BOW	0.4245	0.2512	0.3290	0.1943	0.5054	0.2488	0.2667	0.0702
SD	0.4414	0.2643	0.3400	0.2032	0.5342	0.2688	0.2798	0.0745
WSD (Table 7 in [2])	-	0.2749	-	0.2260	-	0.2946	-	-
DeepTR-BOW (Corpus-specific 300)	0.4430^{b}	0.2591^{b} *(+3.2/-1.9)*	0.3280	0.2103 *(+8.2/+3.5)*	0.5208	0.2646^{b} *(+6.3/-1.6)*	0.2682	0.0718 *(+2.2/-3.6)*
DeepTR-BOW (GOV2 300)	0.4430^{b}	0.2650^{b} *(+5.5/+0.3)*	0.3330	0.2111^{b} *(+8.7/+3.9)*	0.5208	0.2646^{b} *(+6.3/-1.6)*	0.2667	0.0741 *(+5.6/-0.5)*
DeepTR-BOW (ClueWeb09B 300)	0.4454^{b}	0.2657^{b} *(+5.8/+0.5)*	0.3270	0.2129 *(+9.6/+4.8)*	0.5121	0.2685^{b} *(+7.9/-0.1)*	0.2682	0.0718 *(+2.2/-3.6)*
DeepTR-BOW (Google 300)	0.4450^{b}	0.2673^{b} *(+6.4/+1.2)*	0.3380	0.2122^{b} *(+9.3/+4.5)*	0.5221	0.2630^{b} *(+5.7/-2.2)*	0.2667	0.0732 *(+4.2/-1.8)*
DeepTR-SD (Corpus-specific 300)	0.4558^{b}_{s}	0.2754^{b}_{s} *(+9.6/+4.2)*	0.3510	0.2182^{b}_{s} *(+12.3/+7.4)*	0.5490^{b}	0.2831^{b}_{s} *(+13.8/+5.3)*	0.2879^{b}	0.0748 *(+6.5/+0.5)*
DeepTR-SD (GOV2 300)	0.4610^{b}_{s}	0.2781^{b}_{s} *(+10.7/+5.2)*	0.3700^{b}_{s}	0.2223^{b}_{s} *(+14.4/+9.4)*	0.5490^{b}	0.2831^{b}_{s} *(+13.8/+5.3)*	0.2854	0.0806^{b}_{s} *(+14.8/+8.2)*
DeepTR-SD (ClueWeb09B 300)	0.4659^{b}_{s}	0.2810^{b}_{s} *(+11.9/+6.3)*	0.3610^{b}_{s}	0.2279^{b}_{s} *(+17.3/+12.1)*	0.5597^{b}_{s}	0.2890^{b}_{s} *(+16.2/+7.5)*	0.2879^{b}	0.0748 *(+6.5/+0.5)*
DeepTR-SD (Google 300)	0.4627^{b}_{s}	0.2842^{b}_{s} *(+13.1/+7.5)*	0.3560	0.2256^{b}_{s} *(+16.1/+11.0)*	0.5497^{b}	0.2814^{b}_{s} *(+13.1/+4.7)*	0.2869	0.0780^{b}_{s} *(+11.0/+4.7)*

b : Statistically significant difference with BOW
s : Statistically significant difference with SD

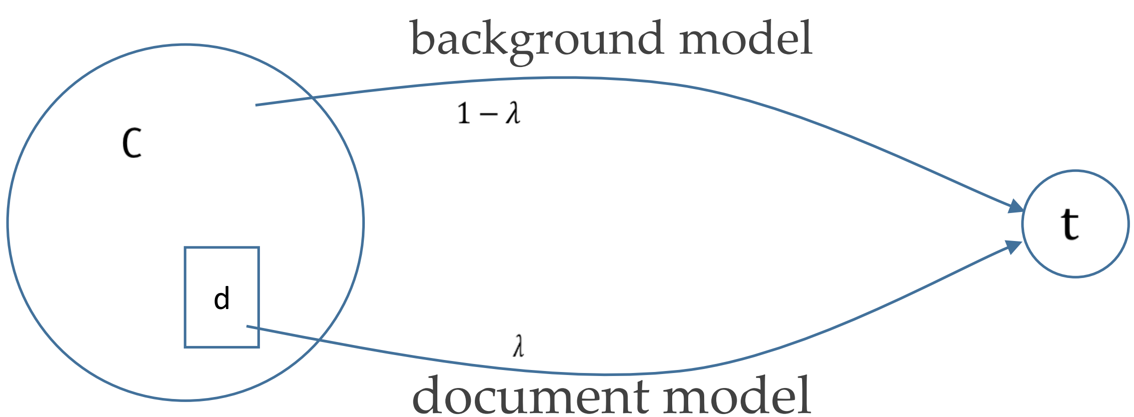
DeepTR term weights perform better than the unweighted query model over all collections
No clear winner among different vector resources

Traditional IR model: Query likelihood

Language modeling approach to IR is quite extensible

$$\mathrm{P}(Q|d) = \prod_{q_i \in Q} P(q_i|d) \qquad \mathrm{P}(t|d) = \lambda P(t|d) + (1-\lambda)P(t|C)$$

- Frequent words in d are more likely (term frequency)
- Smoothing according to the corpus (plays the role of IDF)
- Various ways of dealing with document length normalization

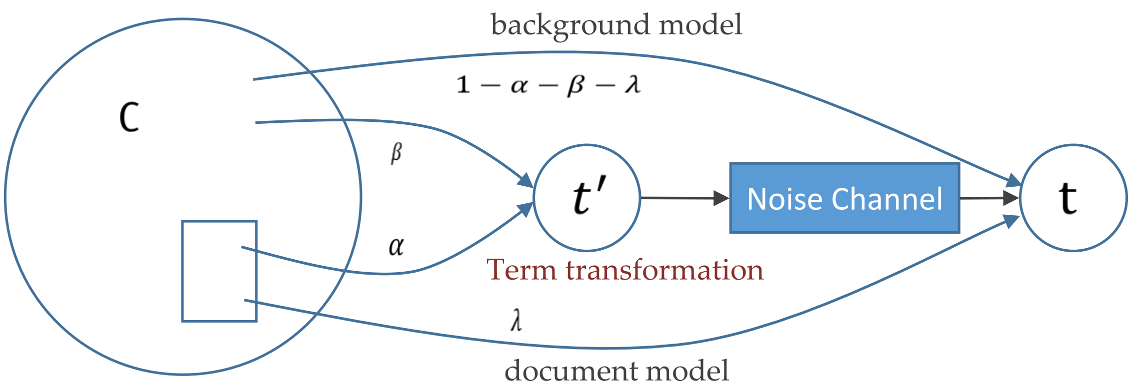


$$P(t|C) = \frac{cf(t)}{cs}$$

$$P(t|d) = \frac{cf(t)}{|d|}$$

A Word Embedding based Generalised Language Model for Information Retrieval, D. Ganguly et. al. 2015 SIGIR.

Generalized Language Model



$$P(t|C) = \frac{cf(t)}{cs}$$

$$P(t|d) = \frac{cf(t)}{|d|}$$

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

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

$$sim(t,t') = \cos(t,t')$$

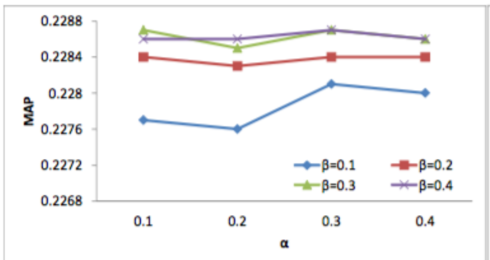
$$\mathrm{P}(Q|d) = \prod_{q_i \in Q} P(q_i|d)$$

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

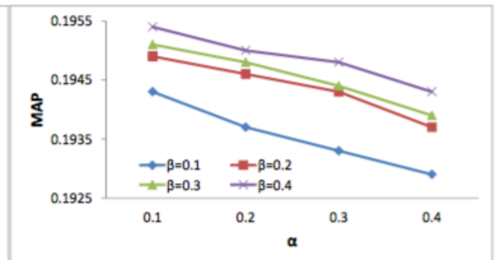
Compares query term with
every document term

A Word Embedding based Generalised Language Model for Information Retrieval, D. Ganguly et. al. 2015 SIGIR.

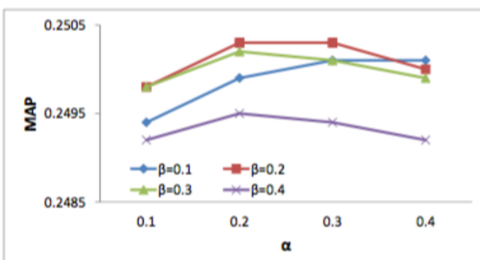
Generalized Language Model



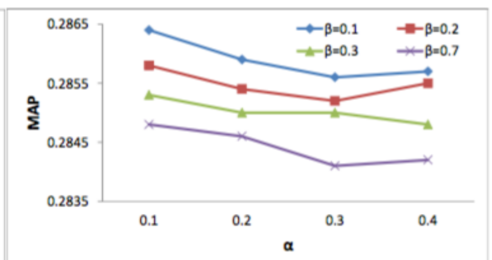
(a) TREC-6



(b) TREC-7



(c) TREC-8



(d) Robust

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

A Word Embedding based Generalised Language Model for Information Retrieval, D. Ganguly et. al. 2015 SIGIR.

Neural Translation Model

NTLM - cbow			
w = insider		w = trading	
u	$p(w\|u)$	u	$p(w\|u)$
insider	0.285	trading	0.216
fraud	0.104	traders	0.103
drexel	0.095	market	0.094
criminal	0.084	stock	0.090
securities	0.084	markets	0.085
racketeering	0.084	futures	0.084

NTLM - skipgram			
w = insider		w = trading	
u	$p(w\|u)$	u	$p(w\|u)$
insider	0.169	trading	0.164
fraud	0.102	traders	0.103
drexel	0.099	futures	0.099
securities	0.096	stock	0.097
racketeering	0.093	exchange	0.094
bribery	0.091	market	0.093

Translation probability from document term u to query term w

$$p_t(w|d) = \sum_{u \in d} \underline{p_t(w|u)}p(u|d)$$

Considers all query-document term pairs

$$p_{cos}(u|w) = \frac{cos(u,w)}{\sum_{u' \in V} cos(u',w)}$$

Based on: Berger and Lafferty. "Information retrieval as statistical translation." SIGIR 1999

Zuccon, Koopman, Bruza, and Azzopardi. "Integrating and evaluating neural word embeddings in information retrieval." Australasian Document Computing Symposium 2015

Neural Translation Model

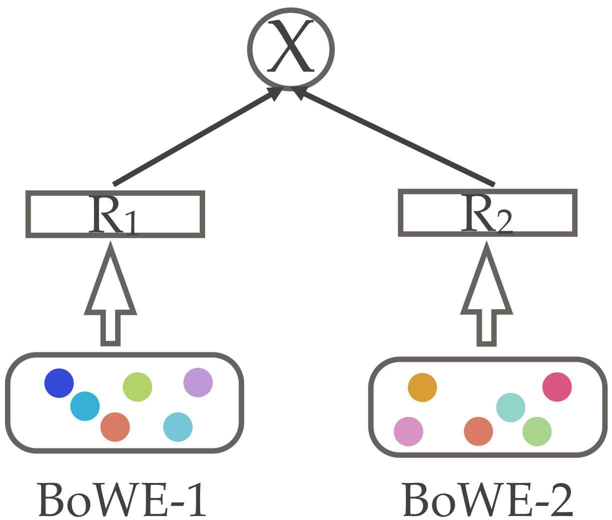
Performance

	AP88-89 ($\mu = 1,000$)		WSJ87-92 ($\mu = 1,500$)		DOTGOV ($\mu = 500$)		MedTrack ($\mu = 3,500$)	
Method	*MAP*	*P@10*	*MAP*	*P@10*	*MAP*	*P@10*	*bpref*	*P@10*
Dirichlet LM	22.69	39.60	21.71	40.80	18.73	24.60	37.69	43.95
TLM-MI	*23.83*^{d}	*41.67*^{d}	20.75	40.73	17.06	22.40	37.02	46.42
TLM-MI-α	22.55	39.73	21.32	40.33	17.15	22.60	37.23	43.70
TLM-MI-s	22.53	39.13	*22.08*	*41.33*	*18.76*	*24.80*	**38.93**	*49.26*^{d}
NTLM-skipgram	**24.27**^{d}	41.00	**22.66**^{d,m}	**42.40**^{d}	**19.32**	**25.00**	38.83	**49.75**^{d}
NTLM-cbow	24.18^{d}	**41.93**^{d}	22.62^{d,m}	42.27^{d}	19.16	24.80	38.77	49.51^{d}

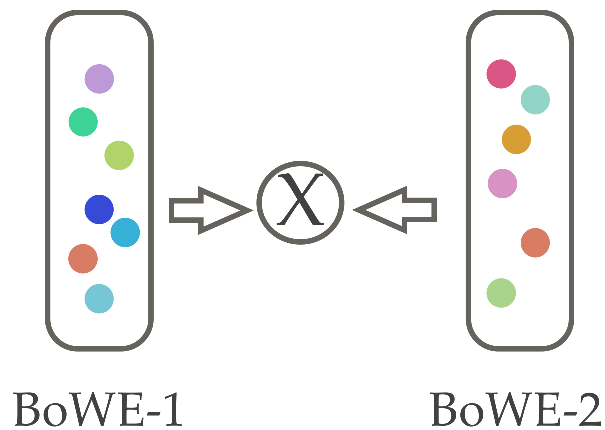
NTLM models provided high quality translations while those of TLM-MI (Translation language model estimated by mutual information) led to poor estimations and consequently losses in retrieval effectiveness.

IR models in the embedding space

- Q: Bag of word vectors
- D: Bag of word vectors
- How to deal with variable length of Q and D?



Composition



Direct Comparison

Composition: Word Embedding-Vector Space

WE-VS score

g

[0.4 1.2 1.6 0.2 1.4 .... 0.7 1.2 4.8 1.5]

[2.6 2.5 1.4 .... 0.9 0.72 1.2]

f_1

f_2

Bag-of-word-embedding Representation

One-hot embedding	idf
[0.1 0.3 0.03 0 0.4 .... 0.05 0.12 0.02]	*4.8*
[0.2 0.13 0.03 0 0 .... 0.07 0.09 0.01]	1.2
[0.13 0.3 0.3 0 0.2 .... 0.08 0.87 0.02]	*1.4*
[0.3 0.4 0.09 0 0.3 .... 0.05 0.34 0.14]	1.6

peace process in the Middle East

Bag-of-word-embedding Representation

One-hot embedding	idf
[0.4 0.3 0.45 0 0.3 .... 0.09 0.24 0.7 0.01]	*1.2*
[0.2 0.1 0.03 0 0.1 .... 0.18 0.91 0.2 0.02]	1.3
[0.1 0.3 0.13 0 0.1 .... 0.16 0.25 0.8 0.03]	*2.5*
...	...

As for the Arabian and Palestinean voices that are against the current negotiations and the so-called peace process, they are not against peace per se, but rather for their well-founded predictions that Israel would NOT give an inch of the West bank (and most probably the same for Golan Heights) back to the Arabs. An 18 months of "negotiations" in Madrid, and Washington proved these predictions. Now many will jump on me saying why are you blaming israelis for no-result negotiations. I would say why would the Arabs stall the negotiations, what do they have to loose ?

$$\vec{Q} = \vec{q_1} + \vec{q_2} + \ldots + \vec{q_m}$$

$$\vec{d} = \mathrm{si}_{w1} \cdot \vec{w_1} + \mathrm{si}_{w2} \cdot \vec{w_2} + \ldots + \mathrm{si}_{w_{|N_d|}} \cdot \overrightarrow{w_{|N_d|}}$$

$$sim(d, Q) = \frac{\vec{d} \cdot \vec{Q}}{|\vec{d}| \cdot |\vec{Q}|}$$

f1 : sum

f2 : weighted sum

g : cosine

Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings, I. Vulic et. al. 2015 SIGIR.

Composition: Word Embedding-Vector Space

	EN→EN			NL→NL		
Model	2001	2002	2003	2001	2002	2003
LM-UNI	.381	.360	.359	.256	.323	.357
LDA-IR	.279	.216	.241	.131	.143	.130
dim:300; *cs*:60						
WE-VS	.324x	.258x	.257y	.203x	.237x	.224x
dim:600; *cs*:60						
WE-VS	.329x	.281x	.262y	.204x	.262x	.231x
LM+LDA	.399	.360	.379	.260	.326	.357
dim:300; *cs*:60						
LM+WE (λ=0.3)	.412y	.381x	.401y	.271x	.349x	.372x
LM+WE (λ=0.5)	.429x	.394x	.407x	.279x	.370x	.382x
LM+WE (λ=0.7)	.451x	.392y	.389	.270	.364x	.373y
dim:600; *cs*:60						
LM+WE (λ=0.3)	.419y	.382x	.403y	.274x	.350x	.373x
LM+WE (λ=0.5)	.436x	.391x	.408x	.282x	.371x	.383x
LM+WE (λ=0.7)	.430x	.392y	.381	.268	.367x	.374y

Too **coarse-grained** compared to LM

Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings, I. Vulic et. al. 2015 SIGIR.

Composition: The Fisher Kernel Framework

FV score

X

g

[0.4 1.2 1.6 0.2 .... 0.7 1.2 4.8 1.5]

[2.6 2.5 1.4 .... 0.9 0.72 1.2]

f_1

f_2

Bag-of-word-embedding Representation

One-hot embedding	idf
[0.1 0.3 0.03 0 0.4 .... 0.05 0.12 0.02]	*4.8*
[0.2 0.13 0.03 0 0 .... 0.07 0.09 0.01]	1.2
[0.13 0.3 0.3 0 0.2 .... 0.08 0.87 0.02]	*1.4*
[0.3 0.4 0.09 0 0.3 .... 0.05 0.34 0.14]	1.6

peace process in the Middle East

Bag-of-word-embedding Representation

One-hot embedding	idf
[0.4 0.3 0.45 0 0.3 .... 0.09 0.24 0.7 0.01]	*1.2*
[0.2 0.1 0.03 0 0.1 .... 0.18 0.91 0.2 0.02]	1.3
[0.1 0.3 0.13 0 0.1 .... 0.16 0.25 0.8 0.03]	*2.5*
[0.7 0.9 0.01 0 0.6 .... 0.15 0.35 0.4 0.26]	*1.3*

Middle East peace processes current status

Fisher Kernel

$$K(X,Y) = G_\lambda^{X'} F_\lambda^{-1} G_\lambda^Y$$

$$= \underbrace{G_\lambda^{X'} L'_\lambda}_{} L_\lambda G_\lambda^Y$$

Fisher Vector

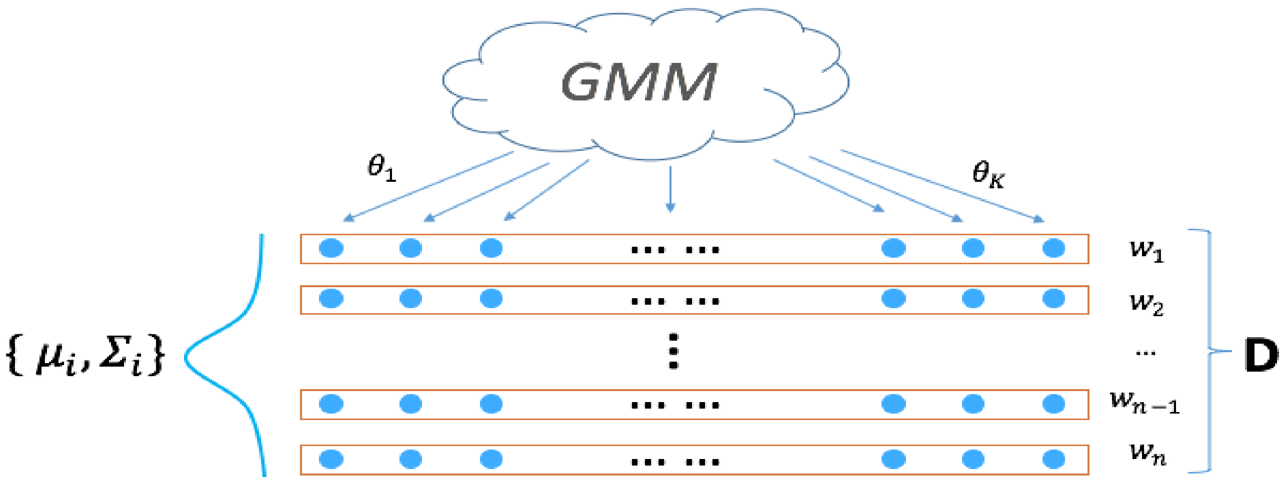
f1 : fisher vector

f2 : fisher vector

g : dot product

Aggregating Continuous Word Embeddings for Information Retrieval, S. Clinching et. al. 2013 ACL.

Composition: The Fisher Kernel Framework



- The word embeddings are assumed to be generated from the Gaussian mixture model(GMM)
- The Fisher Kernel framework: $K(X,Y) = G_\lambda^{X'} F_\lambda^{-1} G_\lambda^{Y} = G_\lambda^{X'} L_\lambda' \underline{L_\lambda G_\lambda^{Y}}$ Fisher Vector
  - ➢ G_λ^X :The gradient vector describes how different model parameters **contribute** to the process of generating the example.
  - ➢ L_λ :The low-rank approximation of the Fisher Information matrix

Composition: The Fisher Kernel Framework

- Learning phase:
  1. Learn an embedding of words in a low-dimensional space
     - $w \longrightarrow E_w = [E_{w,1}, \ldots, E_{w,e}]$
  2. Fit a probabilistic model
     - A mixture model on the word embeddings

- Document representation:
  1. Transfer the BoW representation into a BoWE
     - $\{w_1, \ldots, w_T\} \longrightarrow \{E_{w_1}, \ldots, E_{w_T}\}$
  2. Aggregate the continuous word embeddings E_{wt} using the FK framework

Aggregating Continuous Word Embeddings for Information Retrieval, S. Clinching et. al. 2013 ACL.

Composition: The Fisher Kernel Framework

Collection	Model	ARI	NMI
20NG	PLSA	41.0	57.4
	LDA	40.7	57.9
	LSI	41.0	59.5
	FV	45.2	60.7
TDT	PLSA	64.2	84.5
	LDA	69.4	86.4
	LSI	72.1	88.5
	FV	70.4	88.2

Table 2: Clustering experiments on 20NG and the WebKB TDT Corpus: Mean performance over 20 runs (in %).

Collection	PL2	TFIDF	FV	LSI
CLEF'03	35.7	16.4	23.7	9.2
TREC-1&2	22.6	12.4	10.8	6.5
ROBUST	24.8	12.6	10.5	4.5

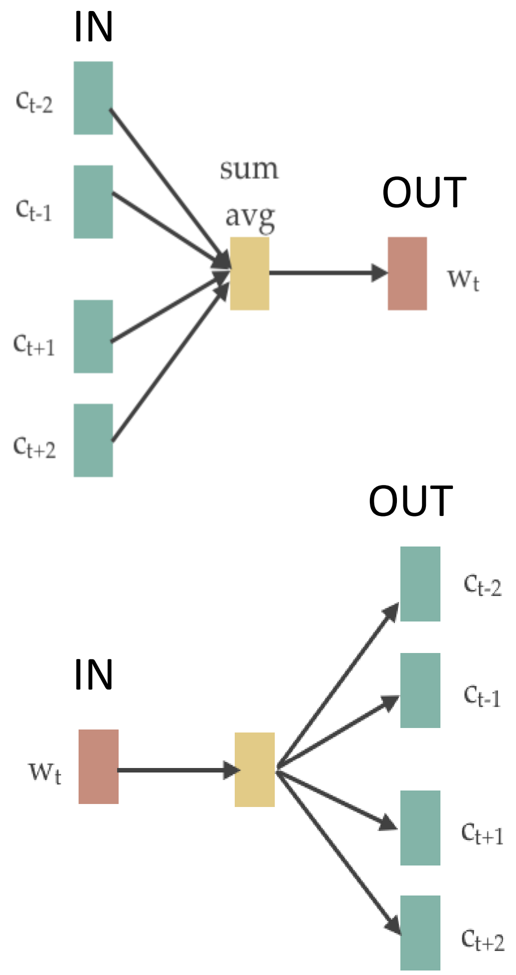
Table 4: Mean Average Precision(%) for the PL2 and TFIDF model on the three IR Collections compared to Fisher Vector and LSI

- FV performs better than the other latent models on document clustering and ad-hoc retrieval.
- There is a significant gap between FV and state-of-the-art IR models.

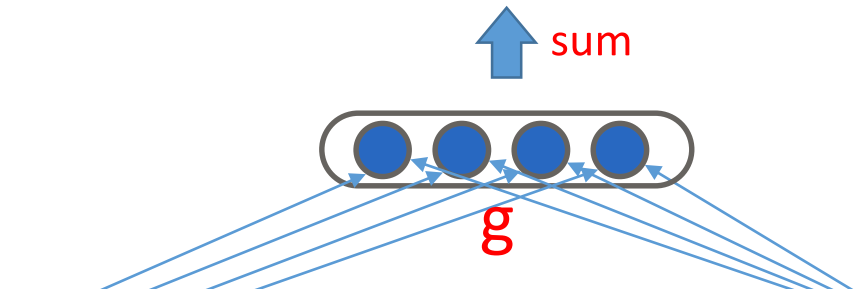
Aggregating Continuous Word Embeddings for Information Retrieval, S. Clinching et. al. 2013 ACL.

Composition: Dual Embedding Space Model

- Two sets of embeddings are trained (W_{IN} and W_{OUT})
- But W_{OUT} is generally discarded
- IN-OUT dot product captures log prob. of co-occurrence



Composition: Dual Embedding Space Model



[2.6 2.5 5.2 0.5 1.4 3.1 .... 0.9 0.72 1.2]

f_2

One-hot embedding	idf
[0.1 0.3 0.03 0 0.4 .... 0.05 0.02]	*4.8*
[0.2 0.13 0.03 0 0 .... 0.07 0.01]	1.2
[0.13 0.3 0.3 0 0.2 .... 0.08 0.02]	*1.4*
[0.3 0.4 0.09 0 0.3 .... 0.05 0.14]	1.6

Bag-of-word-embedding Representation

One-hot embedding	idf
[0.4 0.3 0.45 0 0.3 .... 0.09 0.01]	*1.2*
[0.2 0.1 0.03 0 0.1 .... 0.18 0.02]	1.3
...	...

Bag-of-word-embedding Representation

peace process in the Middle East

As for the Arabian and Palestinean voices that are against the current negotiations and the so-called peace process, they are not against peace per se, but rather for their well-founded predictions that Israel would NOT give an inch of the West bank (and most probably the same for Golan Heights) back to the Arabs. An 18 months of "negotiations" in Madrid, and Washington proved these predictions. Now many will jump on me saying why are you blaming israelis for no-result negotiations. I would say why would the Arabs stall the negotiations, what do they have to loose ?

$$\overline{\mathbf{D}} = \frac{1}{|D|}\sum_{d_j \in D}\frac{d_j}{\|d_j\|}$$

$$DESM(Q, D) = \frac{1}{|Q|}\sum_{q_i \in Q}\frac{q_i^{\top} \cdot \overline{D}}{\|q_i^{\top}\| \cdot \|\overline{D}\|}$$

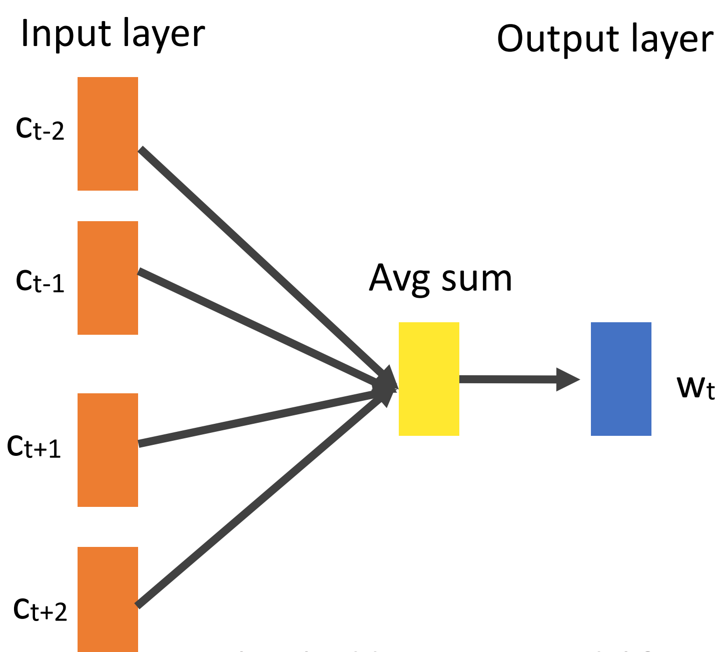
f2 : average sum

g : cosine

A Dual Embedding Space Model for Document Ranking, B. Mitra et. al. 2016 WWW.

Composition: Dual Embedding Space Model

- In-In(or Out-Out) cosine similarities are higher for words that are typically(by type or by function) similar.
- In-Out cosine similarities are higher for words that co-occur often in the train corpus(topically similar).



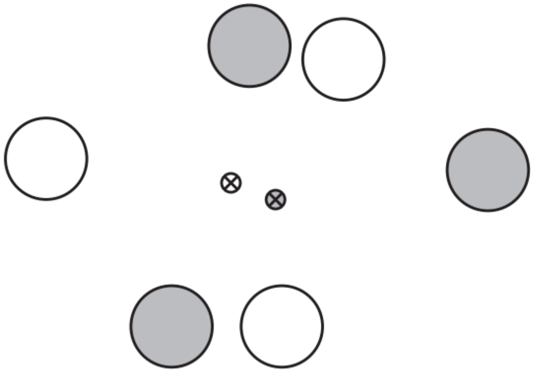
yale		seahawks		eminem	
IN-IN	IN-OUT	IN-IN	IN-OUT	IN-IN	IN-OUT
yale	yale	seahawks	seahawks	eminem	eminem
harvard	faculty	49ers	highlights	rihanna	rap
nyu	alumni	broncos	jerseys	ludacris	featuring
cornell	orientation	packers	tshirts	kanye	tracklist
tulane	haven	nfl	seattle	beyonce	diss
tufts	graduate	steelers	hats	2pac	performs

	Explicitly Judged Test Set		
	NDCG@1	NDCG@3	NDCG@10
BM25	21.44	26.09	37.53
LSA	04.61*	04.63*	04.83*
DESM (IN-IN, trained on body text)	06.69*	06.80*	07.39*
DESM (IN-IN, trained on queries)	05.56*	05.59*	06.03*
DESM (IN-OUT, trained on body text)	01.01*	01.16*	01.58*
DESM (IN-OUT, trained on queries)	00.62*	00.58*	00.81*
BM25 + DESM (IN-IN, trained on body text)	21.53	26.16	37.48
BM25 + DESM (IN-IN, trained on queries)	**21.58**	26.20	37.62
BM25 + DESM (IN-OUT, trained on body text)	21.47	26.18	37.55
BM25 + DESM (IN-OUT, trained on queries)	21.54	**26.42***	**37.86***

A Dual Embedding Space Model for Document Ranking, B. Mitra et. al. 2016 WWW.

Direct Comparison: Comparing short texts

- Weighted semantic network
  - Related to word alignment
  - Each word in longer text is connected to its most similar
  - BM25-like edge weighting
- Generates features for supervised learning of short text similarity



$$f_{sts}(s_l, s_s) = \sum_{w \in s_l} \mathrm{IDF}(w) \cdot \frac{sem(w, s_s) \cdot (k_1 + 1)}{sem(w, s_s) + k_1 \cdot (1 - b + b \cdot \frac{|s_s|}{\mathrm{avgsl}})}$$

$$sem(w, s) = \max_{w' \in s} f_{sem}(w, w').$$

$f_{sem}(w, w')$ returns semantic match score from word embedding

Kenter and de Rijke. Short text similarity with word embeddings. CIKM 2015

Direct Comparison: Comparing short texts

Baseline methods		Acc.	p	r	F1
Convolutional NNs		.699	-	-	.809
VSM		.710	.710	.954	.814
Corpus-based PMI		.726	.747	.891	.813
Our method	**Features**	**Acc.**	**p**	**r**	**F1**
OoB	Unwghtd	.746	.768	.882	.822
OoB	Unwghtd+swsn	0.751	.768	.896	.827
OoB+aux w2v	Unwghtd+swsn	.757	.775	.894	.830
OoB+aux Glv	Unwghtd+swsn	.758	.771	.907	.833
OoB+both aux	Unwghtd+swsn	.766	.781	.906	.839

OoB: Out-of-the-box vectors

Aux: Auxiliary vectors

W2v: Word2vec

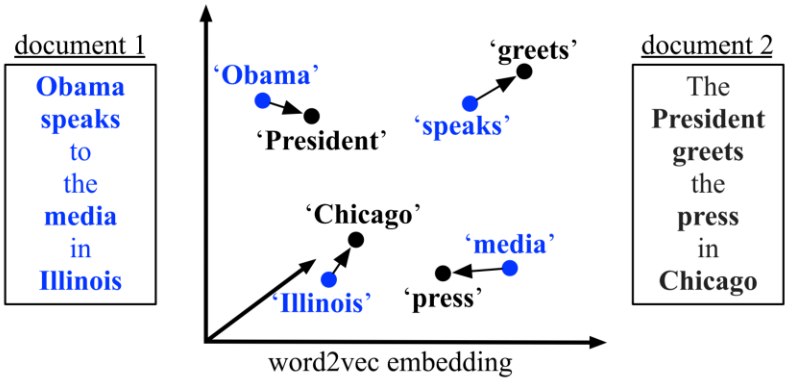
Glv: Glove

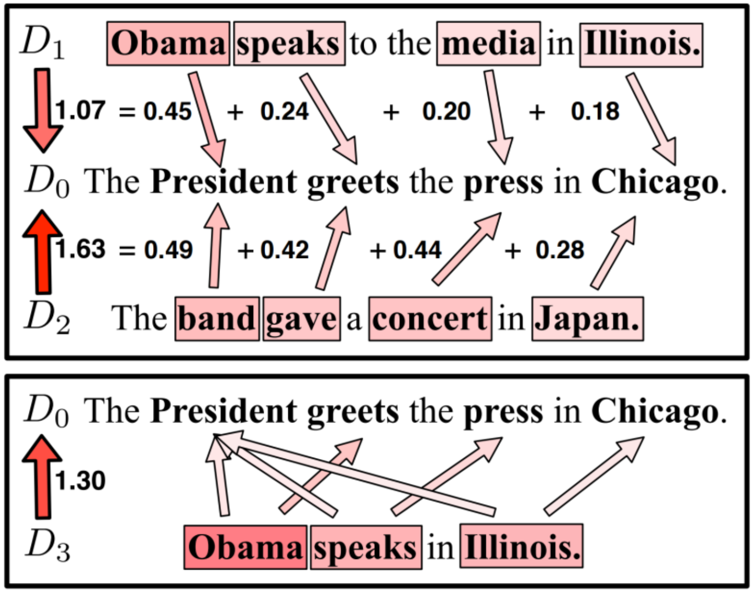
Unwghtd: Unweighted semantic feature

Swsn: Saliency-weighted semantic feature

Kenter and de Rijke. Short text similarity with word embeddings. CIKM 2015

Direct Comparison: Word Mover's Distance





Minimize cost:

$$\min_{\mathbf{T}\geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i,j) \qquad c(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

$$\text{subject to: } \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i \quad \forall i \in \{1, \ldots, n\}$$

$$\sum_{i=1}^{n} \mathbf{T}_{ij} = d'_j \quad \forall j \in \{1, \ldots, n\}.$$

Measure the dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to "travel" to reach the embedded words of another document

Kusner, Sun, Kolkin and Weinberger. From Word Embeddings To Document Distances. ICML 2015
See also: Huang, Guo, Kusner, Sun, Weinberger and Sha. Supervised Word Mover's Distance. NIPS 2016

Direct Comparison: Word Mover's Distance

- Reducing Computation Complexity

Word centroid distance (WCD)

$$\sum_{i,j=1}^{n} T_{ij} c(i,j) = \sum_{i,j=1}^{n} F_{ij} \|x_i - x_j^{'}\|_2$$

$$= \sum_{i,j=1}^{n} \| T_{ij}(x_i - x_j^{'}\|_2 \geq \| \sum_{i,j=1}^{n} T_{ij}(x_i - x_j^{'})\|_2$$

$$= \| \sum_{i=1}^{n} (\sum_{j=1}^{n} T_{ij}) x_i - \sum_{j=1}^{n} (\sum_{i=1}^{n} T_{ij}) x_j^{'} \|_2$$

$$= \| \sum_{i=1}^{n} d_i x_i - \sum_{j=1}^{n} d_j^{'} x_j^{'} \|_2 = \| Xd - Xd^{'} \|_2$$

Relaxed word moving distance (RWMD)

$$\min_{T \geq 0} \sum_{i,j=1} T_{ij} c(i,j)$$

$$\text{subject to}: \sum_{j=1}^{n} T_{ij} = d_i \quad \forall i \in 1, ..., n$$

$$T_{ij}^{*} = \begin{cases} d_i & \text{if } j = \arg\min_j c(i,j) \\ 0 & \text{otherwise} \end{cases}$$

Kusner, Sun, Kolkin and Weinberger. From Word Embeddings To Document Distances. ICML 2015
See also: Huang, Guo, Kusner, Sun, Weinberger and Sha. Supervised Word Mover's Distance. NIPS 2016

Direct Comparison: Word Mover's Distance

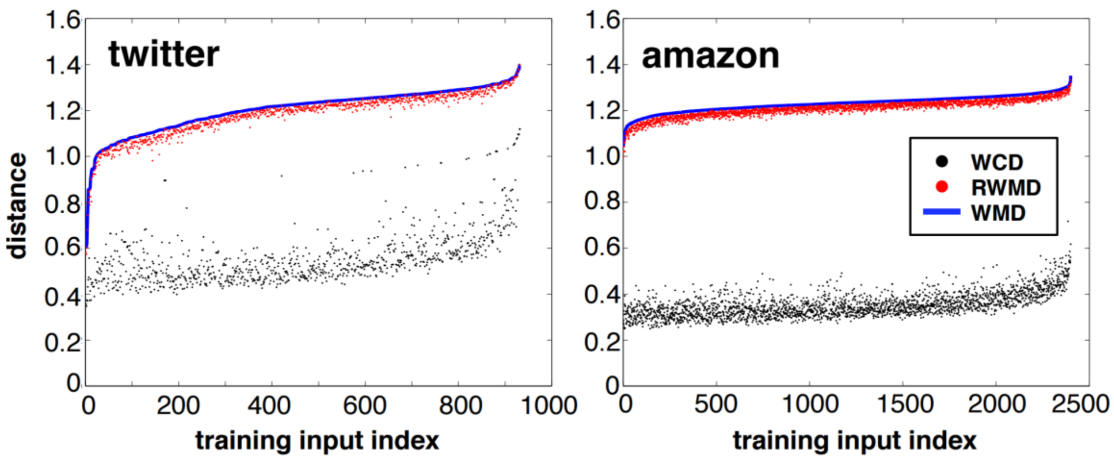


Figure 6. The WCD, RWMD, and WMD distances (sorted by WMD) for a random test query document.

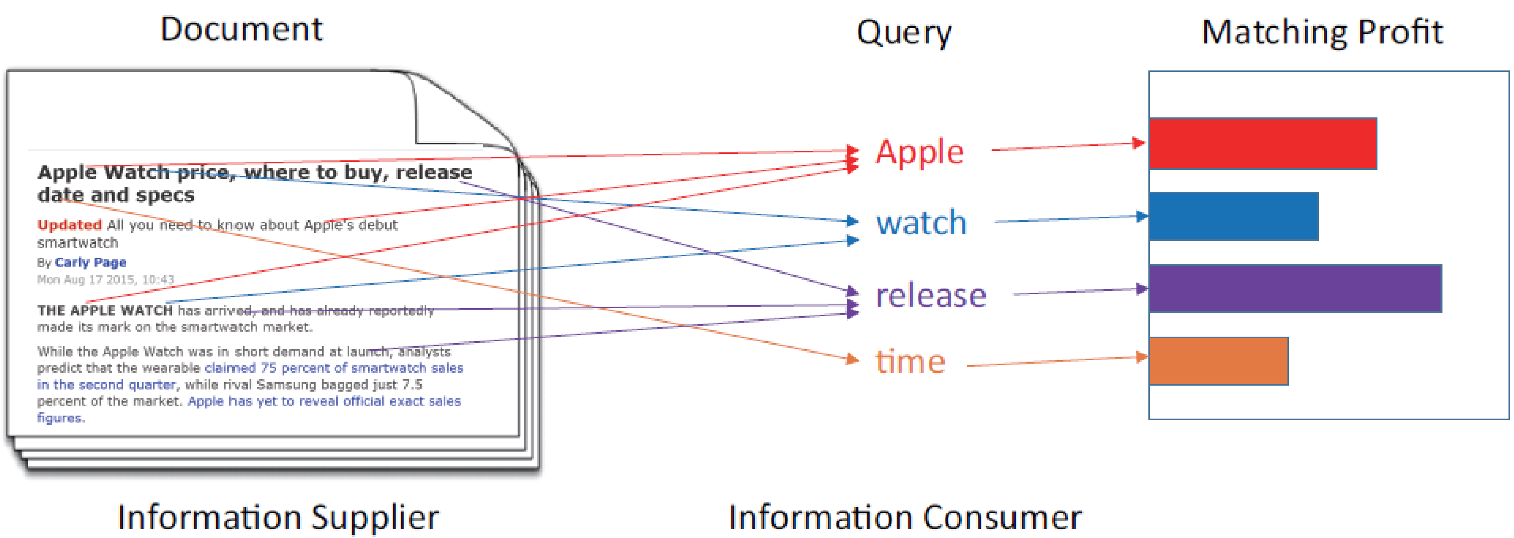
WCD <= RWMD <= WMD
WCD: Word centroid distance
RWMD: Relaxed WMD

Prefetch and prune algorithm:

- Sort by WCD
- Compute WMD on top-k
- Continue, using RWMD to prune 95% of WMD

Kusner, Sun, Kolkin and Weinberger. [From Word Embeddings To Document Distances](). ICML 2015
See also: Huang, Guo, Kusner, Sun, Weinberger and Sha. [Supervised Word Mover's Distance](). NIPS 2016

Direct Comparison: Non-linear Word Transportation



- Matching in IR as a transportation problem
  - The information gain of transporting (i.e., matching) a document word to a query word decides the transportation "profit";
  - The total profit over all the query words defines the relevance between a document and a query;

Semantic Matching by Non-Linear Word Transportation for Information Retrieval, J.F. Guo et. al. 2016 CIKM.

Direct Comparison: Non-linear Word Transportation

- **Semantic Matching as Non-Linear Word Transportation**
  - Given a query and a document with BoWE representations, one aims to find a set of optimal flows $F = \{f_{ij}\}$ that satisfy

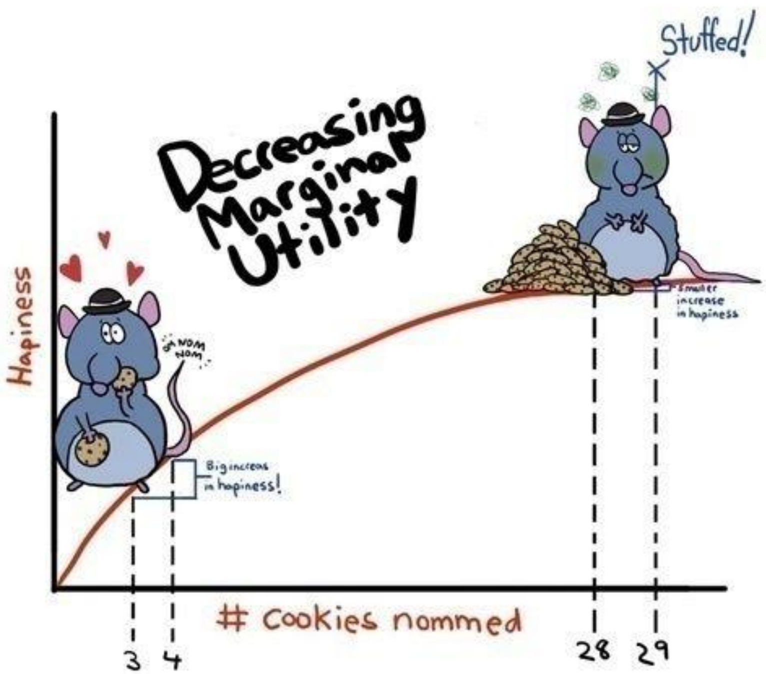
$$\max \quad \sum_{j \in Q} \log \sum_{i \in D} f_{ij} r_{ij}$$

$$\text{subject to:} \quad f_{ij} \geq 0 \qquad \forall i \in D, \forall j \in Q$$

$$\sum_{j \in Q} f_{ij} = c_i \quad \forall i \in D$$

  - No capacity constraint on query side: unlimited capacity to accommodate as much relevant information as possible from the document
  - Non-linear objective function: models diminishing marginal returns on the matching profits

Direct Comparison: Non-linear Word Transportation



- **PIV** (vector space model)

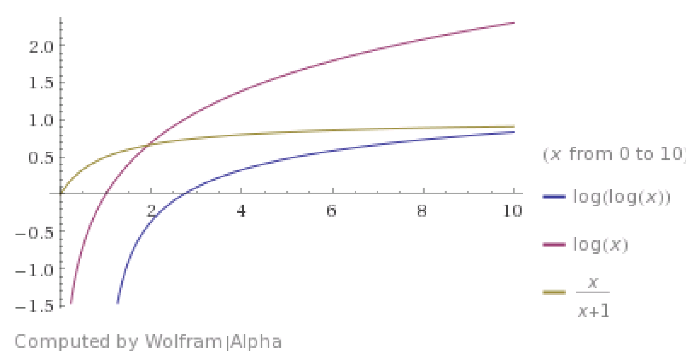
$$\sum_{w\in q\cap d} \frac{\boxed{1+\ln(1+\ln(c(w,d)))}}{(1-s)+s\frac{|d|}{avdl}} \cdot c(w,q) \cdot \ln\frac{N+1}{df(w)}$$

- **DIR** (language modeling approach)

$$\sum_{w\in q\cap d} c(w,q) \times \boxed{\ln(1+\frac{c(w,d)}{\mu \cdot p(w|C)})} + |q| \cdot \ln\frac{\mu}{\mu+|d|}$$

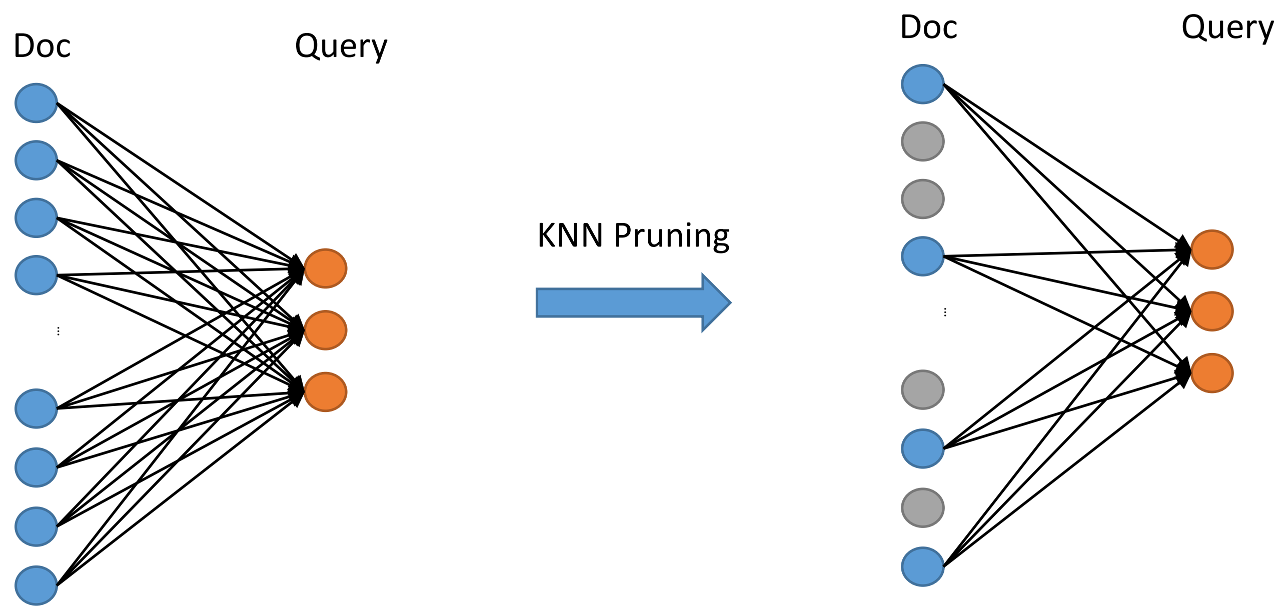
- **BM25** (classic probabilistic model)

$$\sum_{w\in q\cap d} \ln\frac{N-df(w)+0.5}{df(w)+0.5} \times \boxed{\frac{(k_1+1)\times c(w,d)}{k_1((1-b)+b\frac{|d|}{avdl})+c(w,d)}} \times \frac{(k_3+1)\times c(w,q)}{k_3+c(w,q)}$$



Efficient Solution

Efficient pruning and indexing strategy



|V| doc nodes, |Q| query nodes, ~|V|*|Q| edges

Original problem

~K*|Q| doc nodes, |Q| query nodes, ~K*$|Q|^2$ edges

Top-K pruning

Directly solved by convex optimization approaches

Direct Comparison: Non-linear Word Transportation

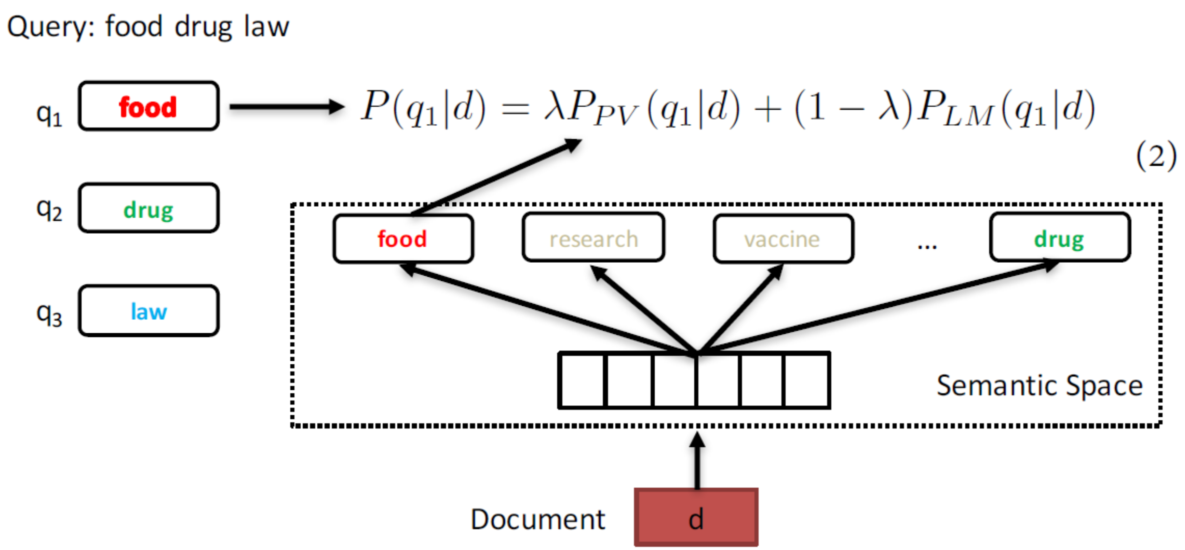
Robust-04 collection

Model Type	Model Name	Topic titles MAP	Topic titles nDCG@20	Topic titles P@20	Topic descriptions MAP	Topic descriptions nDCG@20	Topic descriptions P@20
Exact Matching Baselines	QL	0.253^-	0.415^-	0.369^-	0.246^-	0.391^-	0.334^-
	BM25	0.255^-	0.418	0.370	0.241^-	0.399^-	0.337^-
	SDM	0.263	0.423	0.375	0.261	0.409	0.349
Semantic Matching Baselines	RM3	0.295^+	0.423	0.375	0.264	0.387^-	0.345
	LM+LDA	0.258^-	0.421	0.374	0.247^-	0.392^-	0.336^-
	LM+WE-VS	0.255^-	0.417^-	0.370^-	0.253^-	0.401^-	0.341^-
	WE-GLM	0.255^-	0.417	0.371	0.252^-	0.400^-	0.340^-
Our Approach	NWT	0.274	0.426	0.380	0.268	0.413	0.353

Significant improvement or degradation with respect to NWT is indicated (+/-) (p-value<0:05)

- NWT can significantly outperform basic exact matching baselines;
- NWT even performs better than the state-of-the-art n-gram based model SDM;
- NWT can significantly outperform existing latent models and word embedding based models;
- NWT's performance is comparable with PRF methods;

Direct Comparison: Adapting PV-DBOW for IR



- Improper noise distribution: Negative sampling using IDF instead of corpus frequency
- Overfitting on short documents: L2 regularization constraint on the norm
- Insufficient modeling for word substitution: Predicting context words

Ai, Yang, Guo and Croft. Analysis of the Paragraph Vector Model for Information Retrieval. ICTIR 2016

Direct Comparison: Adapting PV-DBOW for IR

Table 2: Comparison of different models over Robust04 and GOV2 collection. *, + means significant difference over QL, LDA-LM respectively at 0.05 significance level measured by Fisher randomization test. The best performance is highlighted in boldface.

	Robust04 collection					
	Topic titles			Topic descriptions		
Method	MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
QL	0.253	0.415	0.369	0.246	0.391	0.334
LDA-LM	0.258^{*}	0.421	0.374^{*}	0.247	0.392	0.336
PV-LM	0.259^{*}	0.418	0.371	0.247	0.392	0.335
EPV-R-LM	0.259^{*}	0.418	0.370	0.247	0.393	0.336
EPV-DR-LM	0.262^{*}	0.418	0.368	0.252^{*+}	0.397^{*}	0.338^{*}
EPV-DRJ-LM	$\mathbf{0.267}^{*+}$	$\mathbf{0.425}^{*}$	$\mathbf{0.376}^{*}$	$\mathbf{0.253}^{*+}$	$\mathbf{0.404}^{*+}$	$\mathbf{0.347}^{*+}$
	GOV2 collection					
	Topic titles			Topic descriptions		
Method	MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
QL	0.295^{+}	0.409	0.510^{+}	0.249^{+}	0.371	0.470
LDA-LM	0.290	0.406	0.505	0.245	**0.376**	0.468
PV-LM	0.294	0.409	0.510^{+}	0.246	0.364	0.463
EPV-R-LM	0.295^{+}	0.410	0.511^{+}	0.250^{+}	0.368	0.467
EPV-DR-LM	0.296^{+}	0.412	0.512	0.250^{+}	0.371	0.470
EPV-DRJ-LM	$\mathbf{0.297}^{+}$	$\mathbf{0.415}^{*+}$	$\mathbf{0.519}^{*+}$	$\mathbf{0.252}^{*+}$	0.371	**0.472**

All the strategies help improve the embedding based language model

Ai, Yang, Guo and Croft. Analysis of the Paragraph Vector Model for Information Retrieval. ICTIR 2016

3. Query expansion

- Both passages have the same number of gold query matches.
- Yet non-query green matches can be a good evidence of *aboutness*.
- We need methods to consider non-query terms. Traditionally: automatic query expansion.

Query: Albuquerque

Albuquerque is the most populous city in the U.S. state of New Mexico. The high-altitude city serves as the county seat of Bernalillo County, and it is situated in the central part of the state, straddling the Rio Grande. The city population is 557,169 as of the July 1, 2014, population estimate from the United States Census Bureau, and ranks as the 32nd-largest city in the U.S. The Metropolitan Statistical Area (or MSA) has a population of 902,797 according to the United States Census Bureau's most recently available estimate for July 1, 2013.

(a)

Allen suggested that they could program a BASIC interpreter for the device; after a call from Gates claiming to have a working interpreter, MITS requested a demonstration. Since they didn't actually have one, Allen worked on a simulator for the Altair while Gates developed the interpreter. Although they developed the interpreter on a simulator and not the actual device, the interpreter worked flawlessly when they demonstrated the interpreter to MITS in Albuquerque, New Mexico in March 1975; MITS agreed to distribute it, marketing it as Altair BASIC.

(b)

Example from [Mitra et al., 2016]

Query expansion using w2v

- Identify expansion terms using w2v cosine similarity
- 3 different strategies: pre-retrieval, post-retrieval, and pre-retrieval incremental

$$P(w|Q_{exp}) = \alpha P(w|Q) \; + \; (1-\alpha)\frac{Sim(w,Q)}{\sum_{w \in Q_{exp}} Sim(w,Q)}$$

$$Sim(t,Q) = \frac{1}{|Q|} \sum_{q_i \in Q} \mathbf{t}.\mathbf{q_i}$$

where *Qexp* is the set of top K terms from C, the set of candidate expansion terms

Roy, Paul, Mitra and Garain. Using Word embeddings for automatic query expansion. Neu-IR Workshop 2016

Query expansion using w2v

- Performance

Query	Method	Parameters			Metrics		
		K	#fdbck-docs	α	MAP	GMAP	P@5
TREC 6	LM	-	-	-	0.2303	0.0875	0.3920
	Pre-ret	-	100	0.55	0.2406*	0.1026	0.4000
	Post-ret	30	110	0.6	0.2393	0.1028	0.4000
	Increm.	-	90	0.55	0.2354	0.0991	0.4160
	RM3	30	70	-	$0.2634^{k,p,i}$	0.0957	0.4360
TREC 7	LM	-	-	-	0.1750	0.0828	0.4080
	Pre-ret	-	120	0.6	0.1806	0.0956	0.4000
	Post-ret	30	120	0.6	0.1806*	0.0956	0.4280
	Increm.	-	70	0.55	0.1887*	0.1026	0.4360
	RM3	20	70	-	$0.2151^{k,p,i}$	0.1038	0.4160
TREC 8	LM	-	-	-	0.2373	0.1318	0.4320
	Pre-ret	-	120	0.65	0.2535*	0.1533	0.4680
	Post-ret	30	90	0.65	0.2531*	0.1529	0.4600
	Increm.	-	120	0.65	0.2567*	0.1560	0.4680
	RM3	20	70	-	$0.2701^{k,p,i}$	0.1543	0.4760

Query	Method	Parameters			Metrics		
		K	#fdbck-docs	α	MAP	GMAP	P@5
Robust	LM	-	-	-	0.2651	0.1710	0.4424
	Pre-ret	-	90	0.65	0.2842*	0.1869	0.4949
	Post-ret	30	100	0.6	0.2885*	0.1901	0.5010
	Increm.	-	90	0.6	0.2956*	0.1972	0.5051
	RM3	20	70	-	$0.3304^{k,p,i}$	0.2177	0.4949
WT10G	LM	-	-	-	0.1454	0.0566	0.2525
	Pre-ret	-	80	0.6	0.1718*	0.0745	0.2929
	Post-ret	30	90	0.6	0.1709*	0.0769	0.3071
	Increm.	-	100	0.55	0.1724*	0.0785	0.3253
	RM3	20	70	-	$0.1915^{k,p,i}$	0.0782	0.3273

No significant difference between pre- and post- methods

Beats no expansion, but does not beat non-neural expansion

Roy, Paul, Mitra and Garain. [Using Word embeddings for automatic query expansion](). Neu-IR Workshop 2016

Query expansion using w2v

1. Embedding-based Query Expansion
   - Conditional Independence of query term (multiplicative model)

$$p(w|\theta_Q) = \frac{p(\theta_Q|w)p(w)}{p(Q)} \propto p(\theta_Q|w)p(w) \qquad p(q_i|w) = \frac{\delta(q_i, w)}{\sum_{w' \in V} \delta(w', w)}$$

   - Query-Independent Term similarities (mixture model)

$$p(w|\theta_Q) = \sum_{w' \in V} p(w, w'|\theta_Q) = \sum_{w' \in V} p(w|w', \theta_Q)p(w'|\theta_Q) \quad p(w|\theta_Q) \propto \sum_{w' \in Q} \frac{\delta(w, w')}{\sum_{w'' \in V} \delta(w'', w')} \times \frac{c(w', Q)}{|Q|}$$

   - Distance δ has a sigmoid transform to keep a small similar list

2. Embedding-based Relevance Model

$$p(w|\theta_F) \propto \sum_{D \in F} p(w, Q, D) = \sum_{D \in F} p(Q|w, D)p(w|D)p(D))$$

$$p(Q|w, D) = \beta \; p_{tm}(Q|w, D) + (1 - \beta) \; p_{sem}(Q|w, D)$$

$$p_{sem}(Q|w, D) = \prod_{i=1}^{k} p_{sem}(q_i|w, D) \triangleq \prod_{i=1}^{k} \frac{\delta(q_i, w)c(q_i, D)}{Z}$$

Zamani and Croft. Embedding-based query language models. *International Conference on the Theory of Information Retrieval 2016*

Lavrenko and Croft. Relevance based language models. SIGIR 2001

Query expansion using w2v

- performance

Dataset	Metric	MLE	MLE+RM1 (RM3)	EQE1+RM1	EQE2+RM1	MLE+ERM	EQE1+ERM	EQE2+ERM
AP	MAP	0.2236	0.3051	0.3118[12]	0.3115[12]	0.3102[12]	**0.3178**[12]	0.3140[12]
	P@5	0.4260	0.4644	0.4808	0.4795	0.4699	**0.4822**	0.4644
	P@10	0.4014	0.4500	0.4500	0.4452	0.4521	**0.4568**	0.4479
	RI	–	0.47	0.45	0.41	**0.52**	0.47	**0.52**
Robust	MAP	0.2190	0.2677	0.2712[12]	0.2710[12]	0.2711[12]	0.2731[12]	**0.2750**[12]
	P@5	0.4606	0.4581	0.4747	0.4722	0.4639	**0.4797**	0.4730
	P@10	0.3979	0.4191	0.4241	0.4295	0.4241	0.4307	**0.4369**
	RI	–	0.31	**0.39**	0.35	0.31	0.32	0.36
GOV2	MAP	0.2696	0.2938	0.2987[12]	0.2922[1]	0.3005[12]	**0.3012**[12]	0.2957[1]
	P@5	0.5592	0.5592	0.5687	0.5673	0.5823	**0.5850**	0.5782
	P@10	0.5531	0.5599	0.5816	0.5714	0.5830	**0.5844**	0.5782
	RI	–	0.15	**0.22**	0.14	**0.22**	0.20	0.20

Can beat non-neural expansion

Multiplicative better than mixture

Zamani and Croft. Embedding-based query language models. *International Conference on the Theory of Information Retrieval 2016*

Lavrenko and Croft. Relevance based language models. SIGIR 2001

Optimizing the query vector

- Estimating query embedding vectors:
  - ➢ The objective function (optimize towards QL)

$$\vec{q}^* = \arg\max_{\vec{q}} \sum_{w \in V} p(w|\theta_q) \log p(w|\vec{q})$$

$$\arg\max_{\vec{q}} \sum_{w \in V} p(w|\theta_q) \log \delta(\vec{w}, \vec{q})$$

  - ➢ Evaluation via Query Expansion

$$p(w|\theta_q^*) = \alpha \underbrace{p_{mle}(w|\theta_q)}_{\text{MLE of original query}} + (1-\alpha) \underbrace{p(\vec{w}|\vec{q})}_{\text{Query embedding based expansion}}$$

MLE of original query

Query embedding
based expansion

Zamani and Croft. Estimating embedding vectors for queries. *International Conference on the Theory of Information Retrieval 2016*

Optimizing the query vector

- performance

Collection	Metric	QL	MLE+Softmax (AWE)	MLE+Sigmoid	PQV+Softmax	PQV+Sigmoid
AP	MAP	0.2236	0.2470^{0}	0.2486^{0}	0.2695^{012}	$\mathbf{0.2717}^{012}$
	P@5	0.4260	0.4452^{0}	0.4507^{0}	0.4493^{0}	$\mathbf{0.4548}^{01}$
	P@10	0.4014	0.4260^{0}	$\mathbf{0.4274}^{0}$	0.4226^{0}	0.4233^{0}
Robust	MAP	0.2190	0.2299^{0}	0.2303^{0}	0.2355^{012}	$\mathbf{0.2364}^{012}$
	P@5	0.4606	$\mathbf{0.4730}^{0}$	0.4714^{0}	0.4564	0.4591
	P@10	0.3979	0.4237^{0}	$\mathbf{0.4245}^{0}$	0.4083^{0}	0.4141^{0}
GOV2	MAP	0.2696	0.2719	0.2727	0.2771^{012}	$\mathbf{0.2798}^{012}$
	P@5	0.5592	0.5837^{0}	$\mathbf{0.5864}^{0}$	0.5755^{0}	$\mathbf{0.5864}^{0}$
	P@10	0.5531	0.5653^{0}	$\mathbf{0.5721}^{01}$	0.5694^{0}	$\mathbf{0.5721}^{01}$

Pseudo relevance feedback based estimation + sigmoid similarity function work best

Zamani and Croft. Estimating embedding vectors for queries. *International Conference on the Theory of Information Retrieval 2016*

Global vs. local embedding spaces

global	local
cutting	tax
squeeze	deficit
reduce	vote
slash	budget
reduction	reduction
spend	house
lower	bill
halve	plan
soften	spend
freeze	billion

Figure 3: Terms similar to 'cut' for a word2vec model trained on a general news corpus and another trained only on documents related to 'gasoline tax'.

- Train w2v on documents from first round of retrieval
- Fine-grained word sense disambiguation
- A large number of embedding spaces can be cached in practice

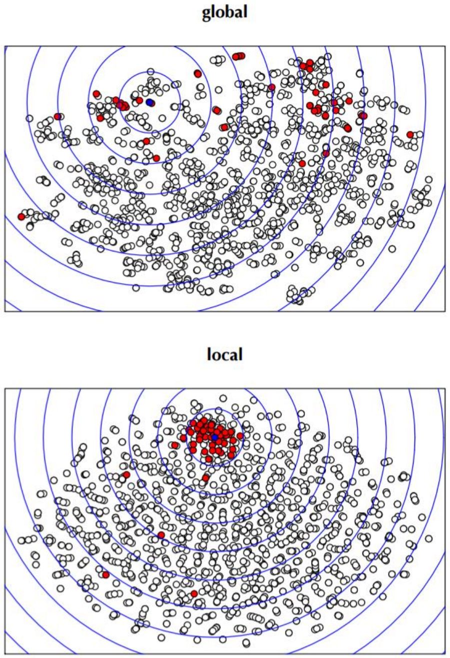


Figure 5: Global versus local embedding of highly relevant terms. Each point represents a candidate expansion term. Red points have high frequency in the relevant set of documents. White points have low or no frequency in the relevant set of documents. The blue point represents the query. Contours indicate distance from the query.

Fernando Diaz at al. Query Expansion with Locally-Trained Word Embeddings. *ACL 2016*

Global vs. local embedding spaces

- Retrieval results of query expansion based on global and local embeddings.

		global						local		
		wiki+giga				gnews	target	target	giga	wiki
	QL	50	100	200	300	300	400	400	400	400
trec12	0.514	0.518	0.518	0.530	0.531	0.530	**0.545**	0.535	**0.563***	0.523
robust	0.467	0.470	0.463	0.469	0.468	**0.472**	0.465	0.475	**0.517***	0.476
web	0.216	0.227	0.229	0.230	**0.232**	0.218	0.216	0.234	0.236	**0.258***

Local embeddings significantly outperform global embeddings for query expansion.

Fernando Diaz at al. Query Expansion with Locally-Trained Word Embeddings. *ACL 2016*

Word Embedding Approaches to IR

Task	Related Work
Ad-hoc Retrieval	ALMasri et al. (2016), Amer et al. (2016), Clinchant and Perronnin (2013), Diaz et al. (2016), GLM (Ganguly et al. (2015)), Mitra et al. (2016), Nalisnick et al. (2016), NLTM (Zuccon et al. (2015)), Rekabsaz et al. (2016), Roy et al. (2016), Zamani and Croft (2016a), Zamani and Croft (2016b), Guo et al. (2016), Zheng and Callan (2015)
Bug Localization	Ye et al. (2016)
Contextual Suggestion	Manotumruksa et al. (2016)
Cross-lingual IR	BWESG (Vulic and Moens (2015))
Detecting Text Reuse	Zhang et al. (2014)
Domain-specific Semantic Similarity	De Vine et al. (2014)
Community Question Answering	Zhou et al. (2015)
Short Text Similarity	Kenter and de Rijke (2015)
Outlier Detection	ParagraphVector (Le and Mikolov (2014))
Sponsored Search	Grbovic et al. (2015b), (Grbovic et al., 2015a)

Summarization

- Word embeddings can be useful for inexact matching
- Embedding based models often perform poorly when applied in isolation, and should be combined with exact matching models (or use telescoping setting).
- These methods seem promising if:
  - ➢ High-quality embeddings/domain-specific embeddings available
  - ➢ No large-scale supervised IR data available
- If large-scale supervised IR data is available … (after the break)

References

- Hinton, G. E., et al. Distributed representations. In Rumelhart, D. E., McClelland, J. L., and PDP Research Group, C., editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1,1986, pages 77–109.
- Huth, A.G., de Heer, W.A., Griffiths, T.L., Theunissen, F.E., Gallant, J.L., 2016. Natural speech reveals the semantic maps that tile human cerebral cortex. Nature 532, 453–458. doi:10.1038/nature17637
- Mikolov, T. et al. Efficient estimation of word representations in vector space. In Proceedings of Workshop of ICLR, 2013.
- Cho, K. Natural Language Understanding with Distributed Representation. 2015.
- Sahlgren, M. (2008). The distributional hypothesis. Italian Journal of Linguistics, 20(1):33–54.
- Sun, F et al. Learning Word Representations by Jointly Modeling Syntagmatic and Paradigmatic Relations. In Proceedings of ACL. 2015, 136–145

- Sahlgren, M. (2008). The distributional hypothesis. Italian Journal of Linguistics, 20(1):33–54.

- Deerwester et al. Indexing by latent semantic analysis. Journal of the American Society for Information Science. 1990, 41(6): 391–407.
- Le, Q. and Mikolov, T. Distributed Representations of Sentences and Documents. ICML 2014, 1188-1196.
- Bengio, Y. et al. A Neural Probabilistic Language Model. JMLR 2003. 1137–1155.
- Morin, F. and Bengio, Y. Hierarchical probabilistic neural network language model. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 2005, 246–252.
- Mnih, A. and Hinton, G. E. A scalable hierarchical distributed language model. Advances in Neural Information Processing Systems 21, 2009, 1081–1088.
- Mnih, A. and Hinton, G. Three new graphical models for statistical language modelling. In Proceedings of ICML, 2007, pages 641–648.

References

- Mikolov, T. et al. Efficient estimation of word representations in vector space. In Proceedings of Workshop of ICLR. 2013.
- Mikolov, T. et al. Distributed representations of words and phrases and their compositionality. In NIPS, 2013, 3111–3119.
- Levy, O. and Goldberg, Y. Neural word embedding as implicit matrix factorization. In NIPS. 2014, 2177–2185.
- Levy, O. and Goldberg, Y. Linguistic regularities in sparse and explicit word representations. CoNLL-2014.
- Levy, O. et al. Improving distributional similarity with lessons learned from word embeddings. TACL 3:211–225, 2015.
- Pennington, J. et al. Glove: Global vectors for word representation. In Proceedings of EMNLP 2014, 1532–1543.
- Lai, S. et.al. How to Generate a Good Word Embedding? arXiv, 2015.
- Neelakantan, A et al. Efficient non-parametric estimation of multiple embeddings per word in vector space. In Proceedings of EMNLP, 2014, 1059–1069.
- Huang, E. H.et al. Improving word representations via global context and multiple word prototypes. In Proceedings of ACL 2012, 873–882.
- Tian, F. et al. A probabilistic model for learning multi-prototype word embeddings. In Proceedings of COLING 2014, 151–160
- Li, J. and Jurafsky, D. Do multi-sense embeddings improve natural language understanding? In Proceedings of EMNLP, 2015, 1722–1732.
- Luong, M.-T. et al. Better word representations with recursive neural networks for morphology. In Proceedings of CoNLL 2013, 104–113.

References

- Qiu, S. et al. Co-learning of word representations and morpheme representations. In Proceedings of COLING 2014 141–150.
- Botha, J. A. and Blunsom, P. Compositional morphology for word representations and language modelling. In Proceedings of ICML. 2014, 1899–1907.
- Sun, F.et al. Inside out: Two jointly predictive models for word representations and phrase representations. In Proceedings of tAAAI. 2016, 2821–2827.
- Yu, M. and Dredze, M. . Improving lexical embeddings with semantic knowledge. In Proceedings of tACL 2014, 545–550.
- Bian, J., et al. Knowledge-powered deep learning for word embedding. Machine Learning and Knowledge Discovery in Databases, volume 8724 of LNCS, 2014132–148.
- Rubinstein, D., et al. How well do distributional models capture different types of semantic knowledge? In Proceedings of ACL, 2015, 726–730.
- Liu, Q., et al. Learning semantic word embeddings based on ordinal knowledge constraints. In Proceedings of ACL, 2015, 1501–1511.
- Faruqui, M., et al. Retrofitting word vectors to semantic lexicons. In Proceedings of NAACL. 2015, 1606–1615.
- [Salton et al. 1975] A theory of term importance in automaLc text analysis. G. Salton, C.S. Yang and C. T. Yu. Journal of the American Society for InformaLon Science, 1975.

References

- [Singhal et al. 1996] Pivoted document length normalizaLon. A. Singhal, C. Buckley and M. Mitra. SIGIR 1996.

- [Harter 1975] A probabilisLc approach to automaLc keyword indexing. S. P. Harter. Journal of the American Society for InformaLon Science, 1975.
- [Robertson&Sparck Jones 1976] Relevance weighLng of search terms. S. Robertson and K. Sparck Jones. Journal of the American Society for InformaLon Science, 1976.
- [Maron&Kuhn 1960] On relevance, probabilisLc indexing and informaLon retrieval. M. E. Maron and J. L. Kuhns. Journal o Éte ACM, 1960.
- [van Rijsbergen 1977] A theoreLcal basis for the use of co=occurrence data in informaLon retrieval. C. J. van Rijbergen. Journal of DocumentaLon, 1977.
- [Robertson&Walker 1994] Some simple effecLve approximaLons to the 2= Poisson model for probabilisLc weighted retrieval. S. E. Robertson and S. Walker. SIGIR 1994.
- [Ponte&Crof 1998] A language modeling approach to informaLon retrieval. J. Ponte and W. B. Crof. SIGIR 1998.
- [Zhai&Lafferty 2001] A study of smoothing methods for language models applied to ad hoc informaLon retrieval. C. Zhai and J. Lafferty. SIGIR 2001.
- [Lavrenko&Crof 2001] Relevance=based language models. V. Lavrenko and B. Crof. SIGIR 2001.
- [Kurland&Lee 2004] Corpus structure, language models, and ad hoc informaLon retrieval. O. Kurland and L. Lee. SIGIR 2004.