Text Mining I: Word Embedding and Spherical Text Embedding

JIAWEI HAN
COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

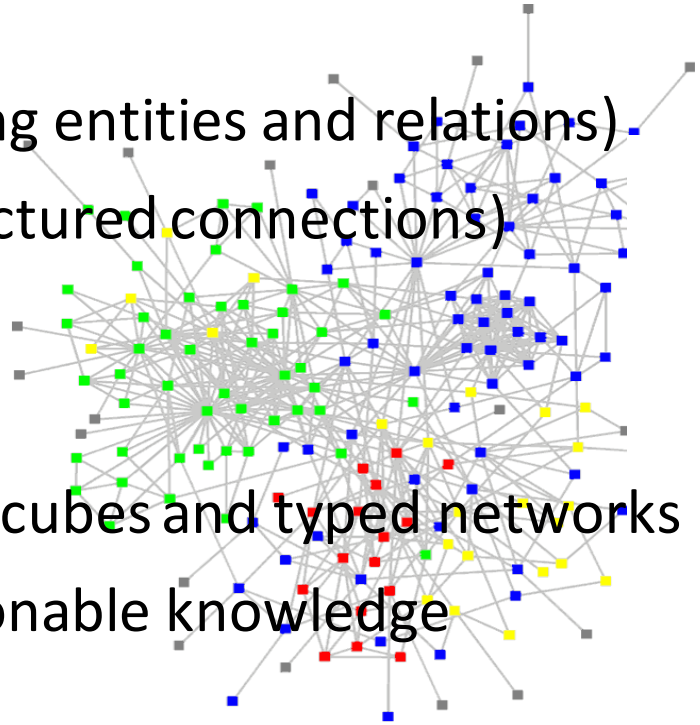
MAY 20, 2023

Outline

- Text Mining, Text Similarity and Text Embedding
- Unsupervised word embedding
  - Context-free representation:
    - Euclidean Embedding: Word2Vec, GloVe, fastText
      - Local-Corpus Based Embedding
    - Hyperbolic Embedding: Poincaré and Lorentz Embedding
    - Spherical Text Embedding: JoSE (Joint Spherical Text Embedding)
  - Contextualized representation: ELMo, BERT, XLNet
- CatE: Category-Name Guided Text Embedding for Topic Mining
- Looking Forward

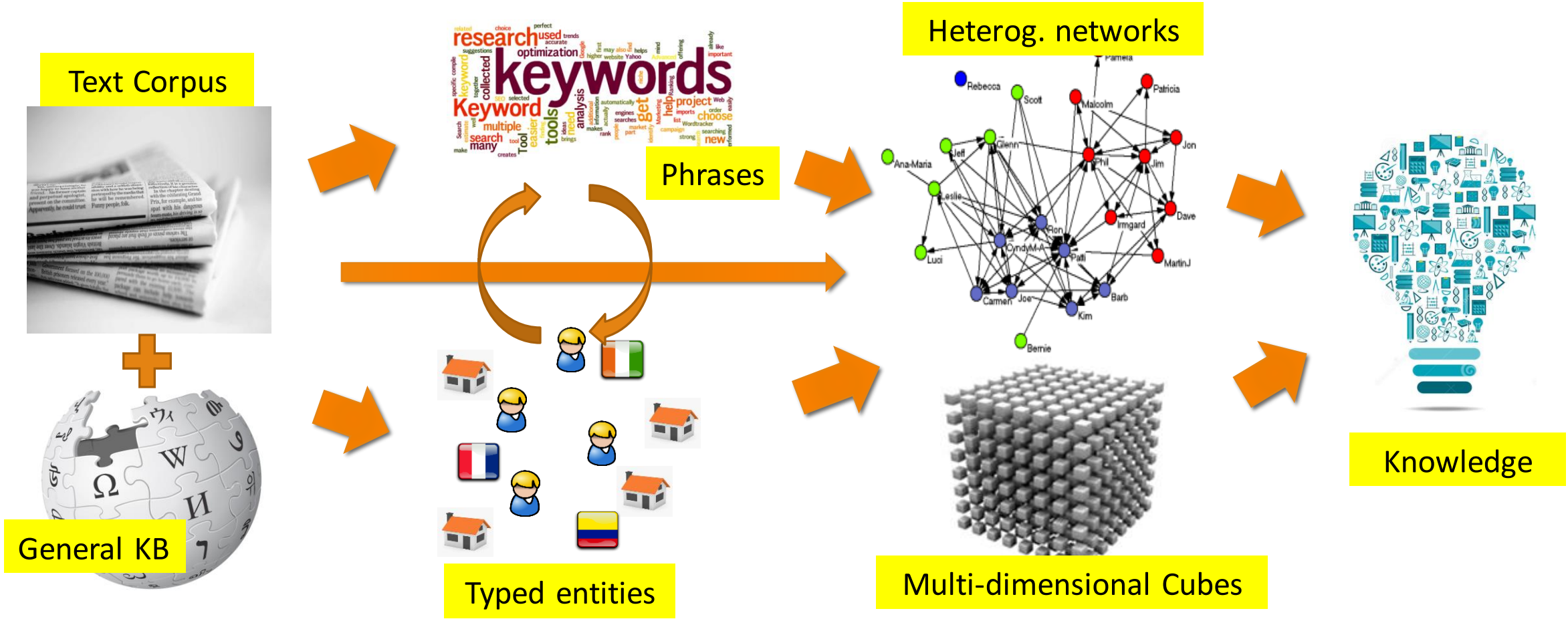
Over 80% of Our Big Data is Unstructured Text Data

- Ubiquity of big unstructured, text data
  - Big Data: Over 80% of our data is from text/natural language/social media, unstructured/semi-structured, noisy, dynamic, …, but inter-related!
- How to mine such big data systematically?
  - Structuring (i.e., transforming unstructured text into structured, typed, interconnected entities/relationships)
  - Embedding (i.e., computing similarities among entities and relations)
  - Networking (take advantage of massive, structured connections)
- Our roadmap:
  - Mining hidden structures from text data
  - Turning text data into multidimensional text-cubes and typed networks
  - Mining cubes and networks to generate actionable knowledge



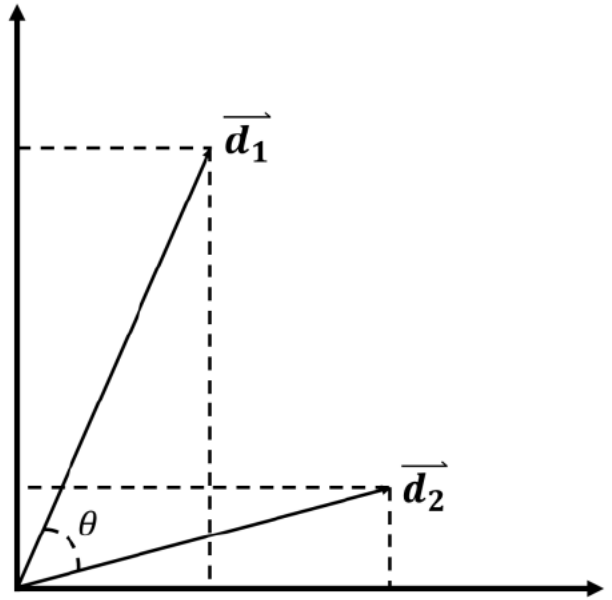
Bottleneck: Mining Unstructured Text for Structures

- One of the most challenging issue at mining big data: structuring and mining text!!
- Bottleneck: How to automatically generate structures from text data?
  - Automated mining of phrases, topics, entities, links and types from text corpora



Text Representation: Vector Space Representation

- Vector Space Representation
  - The sparse, multidimensional representation of text used in most applications
  - Term frequency (TF): $tf_i = N(t_i, d)$ or take the log: $f_i = log\ (tf_i + 1)$
  - Inverted document frequency (IDF): $idf_i = \log\ (N/df_i)$
  - TF-IDF: $tf\_idf_i = tf_i$ X idf_i
    - *Term i appears frequently in the current doc d but infrequent in other docs*
- Normalization (due to different doc length)
  - L-1 normalization: $N(\boldsymbol{w}) = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_1} = \frac{\boldsymbol{w}}{\sum_{i=1}^{V} w_i}$
  - L-2 normalization: $N(\boldsymbol{w}) = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2} = \frac{\boldsymbol{w}}{\sqrt[2]{\sum_i w_i^2}}$
  - L-infinity normalization: $N(\boldsymbol{w}) = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_\infty} = \frac{\boldsymbol{w}}{\max_i\{w_i\}}$



Similarity Computation in Text

- A **document** can be represented by a bag of terms or a long vector, with each attribute recording the *frequency* of a particular term (such as word, keyword, or phrase) in the document

Document	*team*	*coach*	*hockey*	*baseball*	*soccer*	*penalty*	*score*	*win*	*loss*	*season*
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

- *Cosine measure*: If d_1 and d_2 are two vectors (e.g., term-frequency vectors), then

$$cos(d_1, d_2) = \frac{d_1 \bullet d_2}{\| d_1 \| \times \| d_2 \|} \qquad \text{cosine}(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \sqrt{\sum_{i=1}^{d} y_i^2}}$$

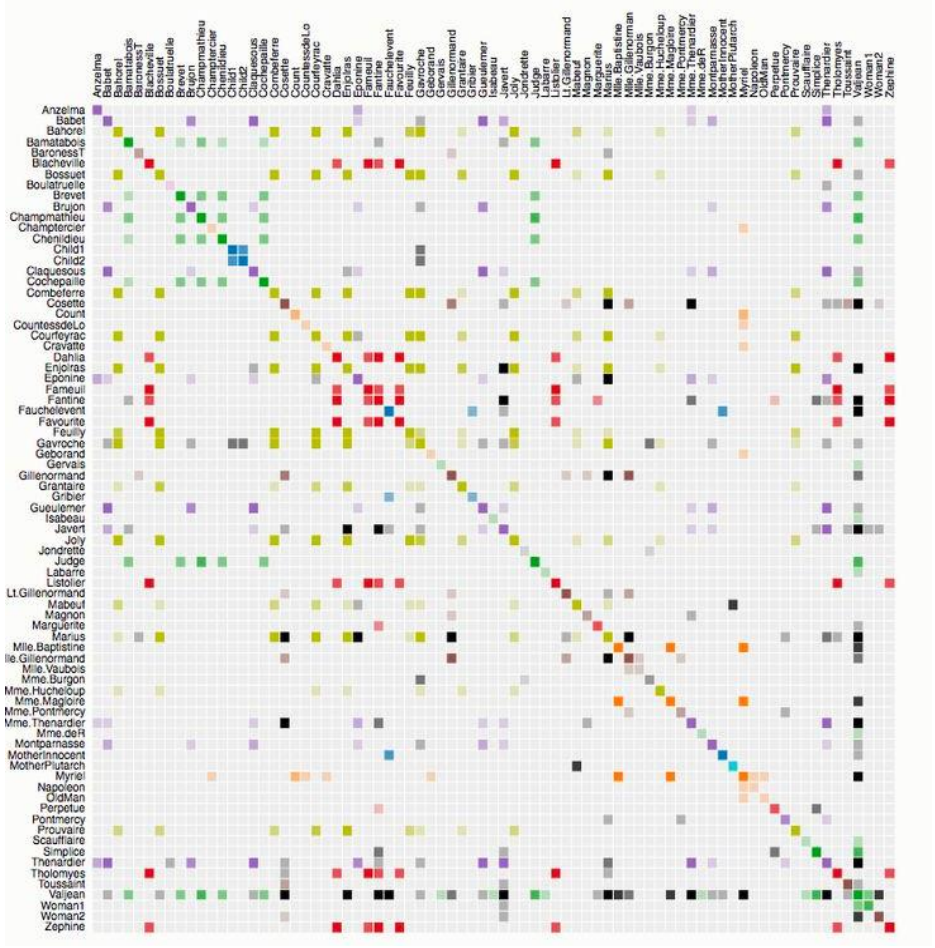
where $\bullet$ indicates vector dot product, $||d||$: the length of vector d

- *Jaccard coefficient is used in some analysis:*

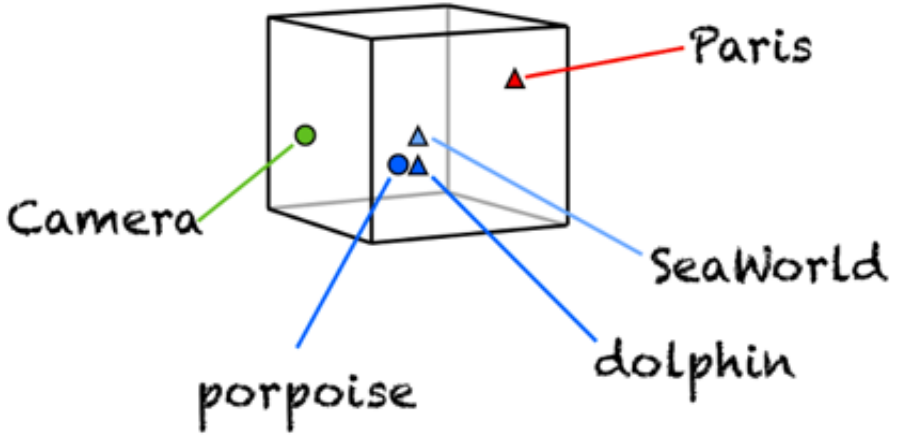
$$\text{Jaccard}(S_x, S_y) = \frac{|S_x \cap S_y|}{|S_x \cup S_y|} \qquad \text{Jaccard}(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{d} x_i \cdot y_i}{\sum_{i=1}^{d} x_i^2 + \sum_{i=1}^{d} y_i^2 - \sum_{i=1}^{d} x_i \cdot y_i}$$

Big Data Challenge: The Curse of High-Dimensionality

- Text: Word co-occurrence statistics matrix



- High-dimensionality:
  - There are over **171k** words in English language
- Redundancy:
  - Many words share similar semantic meanings
    - Sea, ocean, marine..

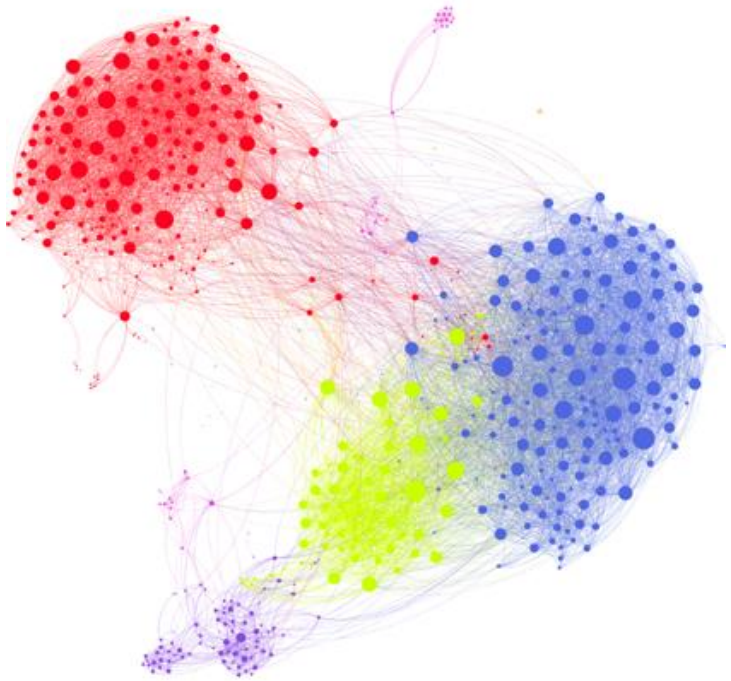


Multi-Genre Network Challenge: High-Dimensional Data too!

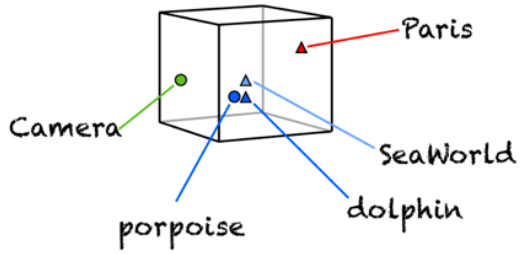
- Adjacency Matrix

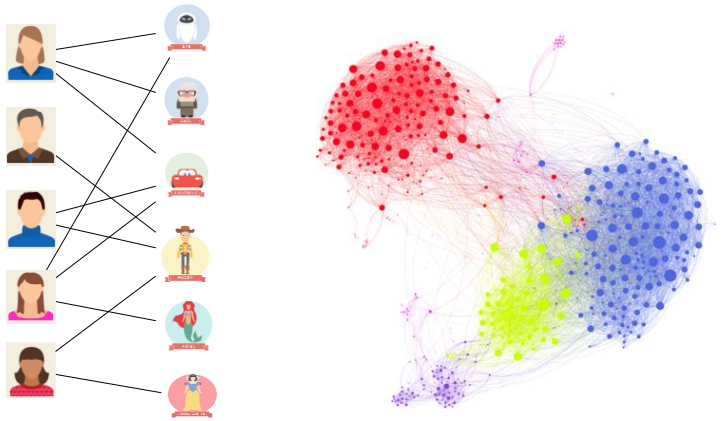
	1	2	3	4	5	6	7	8	9	10	…
1	0	1	1	1	1	0	0	1	0	0	…
2	1	0	1	1	0	0	1	0	0	0	…
3	1	1	0	1	0	0	0	0	1	0	…
4	1	1	1	0	0	0	0	0	0	0	…
5	1	0	0	0	0	0	0	0	0	0	…
6	1	0	0	0	0	0	0	0	0	0	…
7	1	0	0	0	1	0	0	0	0	0	…
8	1	1	1	1	0	0	0	0	0	0	…
9	0	0	1	0	0	0	0	0	0	1	…
10	0	0	1	0	0	0	0	1	0	1	…
11	0	0	0	0	0	0	0	0	1	1	…
12	0	1	0	0	0	0	0	0	1	1	…
13	1	0	0	0	0	0	0	0	1	1	…
14	0	0	1	0	0	1	1	1	0	1	…
15	0	0	0	0	0	1	1	1	1	0	…
…	…	…	…	…	…	…	…	…	…	…	…

- High-dimension:
  - Facebook has 1860 Million monthly active users (Mar. 2017)
- Redundancy:
  - Users in the same cluster are likely to be connected



Solution to Data & Network Challenge: Dimension Reduction

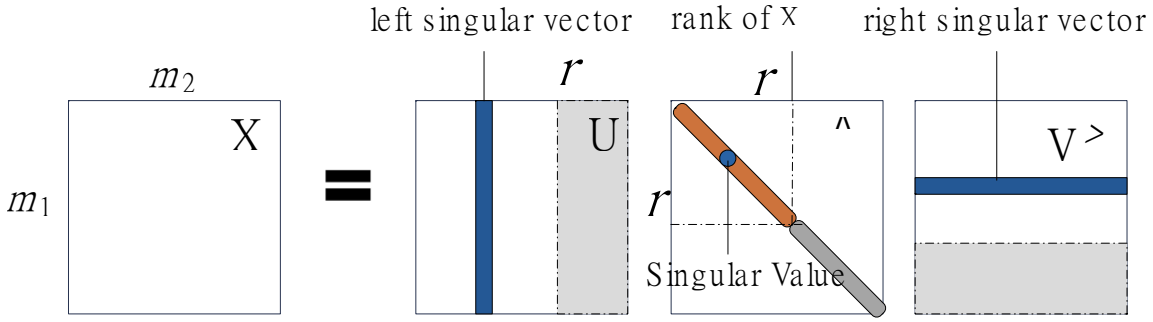




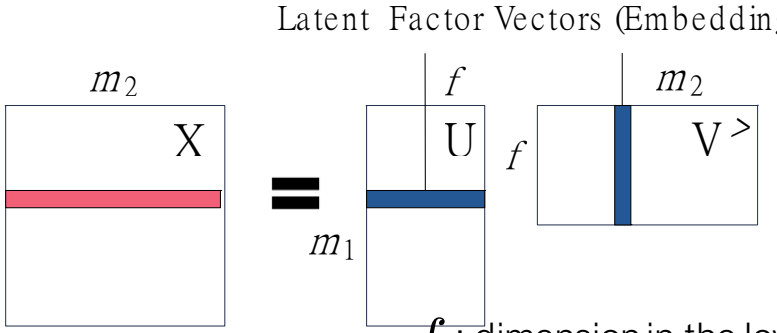
- Why Low-dimensional Space?
  - Visualization
  - Compression
  - Explanatory data analysis
  - Fill in (impute) missing entries (link/node prediction)
  - Classification and clustering
  - Identify / point
- How to automatically identify the lower-dimensional space that the high-dimensional data (approximately) lie in

Dimension Reduction Approaches: Low-rank Estimation vs. Embedding Learning

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$$



$$\mathbf{X} = \mathbf{U}\mathbf{V}^\top$$



f : dimension in the low-dimensional space

- Low-rank estimation
  - Data recovery
  - Imposing low-rank assumption
    - Regularization
  - Low-dimension vector space
    - Singular vectors (U)
    - = r
  - Low-rank Model

- Embedding Learning
  - Representation Learning
  - Project data into a low-dimensional space
  - Low-dimensional vector space
    - Spanned by columns of U
    - ≤ f
  - Generalized Low-rank Model

Outline

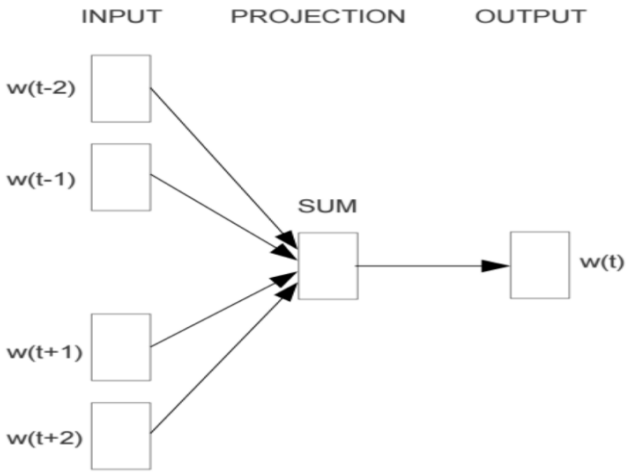
- Text Mining, Text Similarity and Text Embedding
- Unsupervised word embedding
  - Context-free representation:
    - Euclidean Embedding: Word2Vec, GloVe, fastText
      - Local-Corpus Based Embedding
    - Hyperbolic Embedding: Poincaré and Lorentz Embedding
    - Spherical Text Embedding: JoSE (Joint Spherical Text Embedding)
  - Contextualized representation: ELMo, BERT, XLNet
- CatE: Category-Name Guided Text Embedding for Topic Mining
- Looking Forward

Word2Vec and Word Embedding

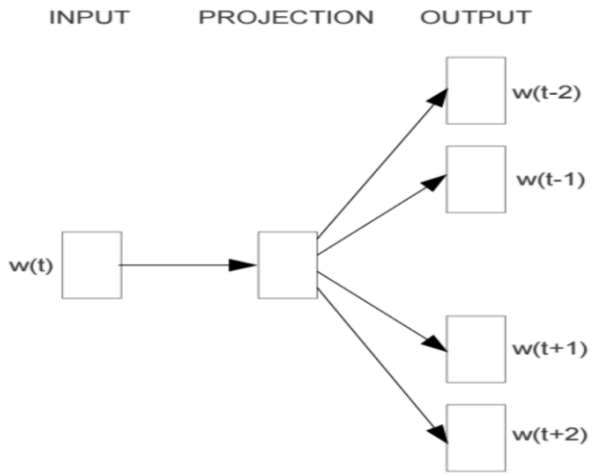
- Word2vec: A two-layer neural net that processes text
  - Proposed by T. Mikolov et al. at Google (2013)
  - Input: A large text corpus
  - Output: A set of vectors, feature vectors for words in that corpus, of 10^2 dimensions
- Words sharing common contexts are embedded in close proximity in the vector space
  - Embedding vectors created by Word2vec: better than LSA (Latent Semantic Analysis)
  - E.g., Sweden → Norway, Denmark, Finland, Switzerland, Belgium, Netherlands, …
- Given enough data, usage and contexts, Word2vec can make highly accurate guesses about a word's meaning based on past appearances

Word2Vec: CBOW vs. Skip-Gram Models

- Two model architectures: Given a set of sentences (also called **corpus**),
  - Continuous bag-of-words (CBOW): Uses the contexts to predict the current word
    - *faster than skip-gram, slightly better accuracy for the frequent words*
  - Skip-gram: Uses the current word to predict its neighbors (its context)
    - Weigh nearby context words more heavily than more distant context words
    - *Works well with small amount of the training data, represents well even rare words or phrases*



CBOW



Skip-gram

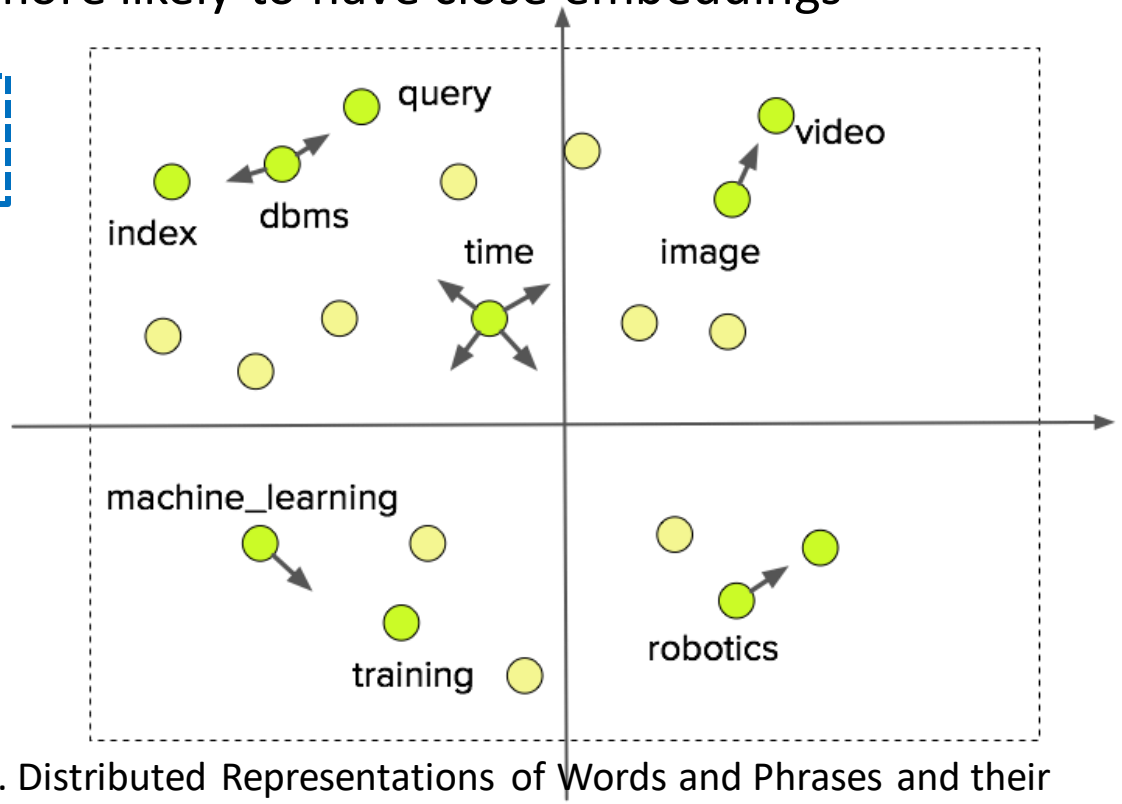
Unsupervised Word Embedding: Word2Vec

- Unsupervised word embedding learning pushes together terms that share same or similar contexts
- Specifically, Word2Vec maximizes the probability of observing a word based on its contexts
- As a result, semantically coherent terms are more likely to have close embeddings

Co-occurred words in a local context window

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

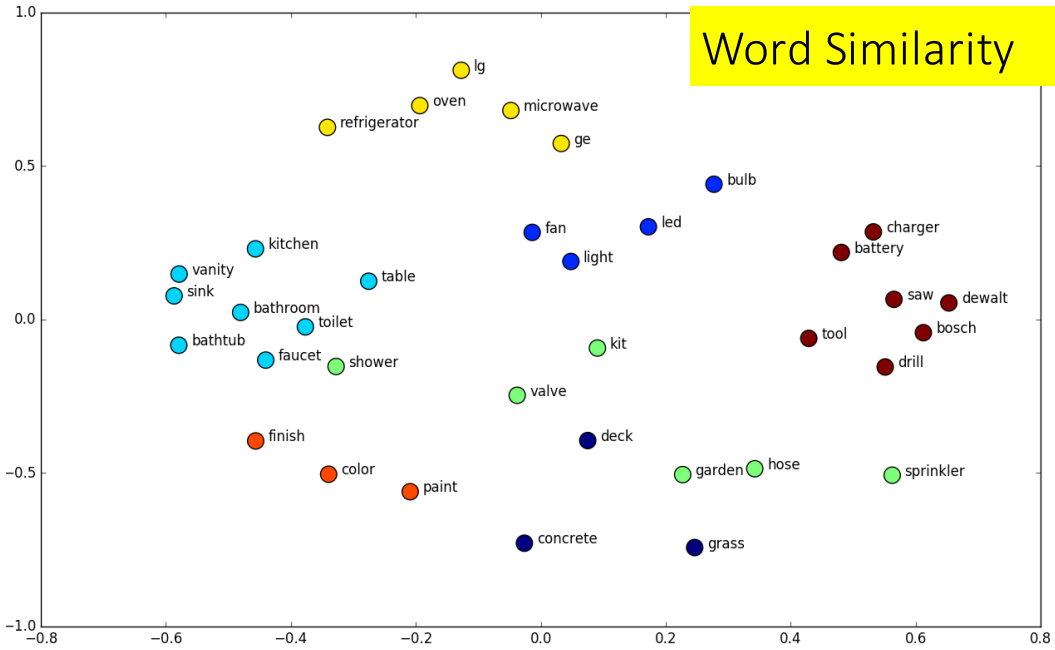
$$p(w_O|w_I)=\frac{\exp\left({v'_{w_O}}^{\top}v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top}v_{w_I}\right)}$$

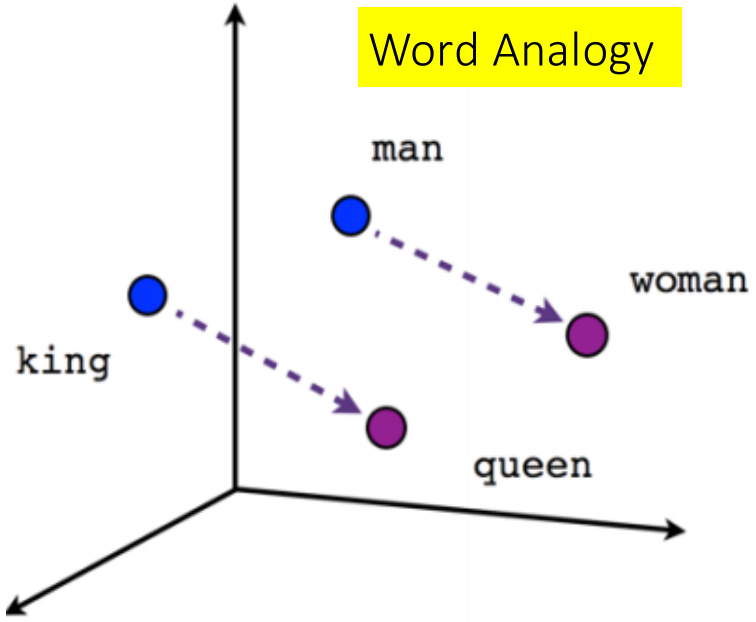


Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. NIPS.

Text Embedding: A Milestone in NLP and ML

- A milestone in NLP and ML: Unsupervised learning of text representations
- Embed one-hot vectors into lower-dimens. space—Address "curse of dimensionality"
- Word embedding captures useful properties of word semantics
  - Word similarity: Words with similar meanings are embedded closer
  - Word analogy: Linear relationships between words (e.g., king − queen = man−woman)





Typical embedding methods:
- Word2Vec
- GloVe
- fastText

Trained in Euclidean space

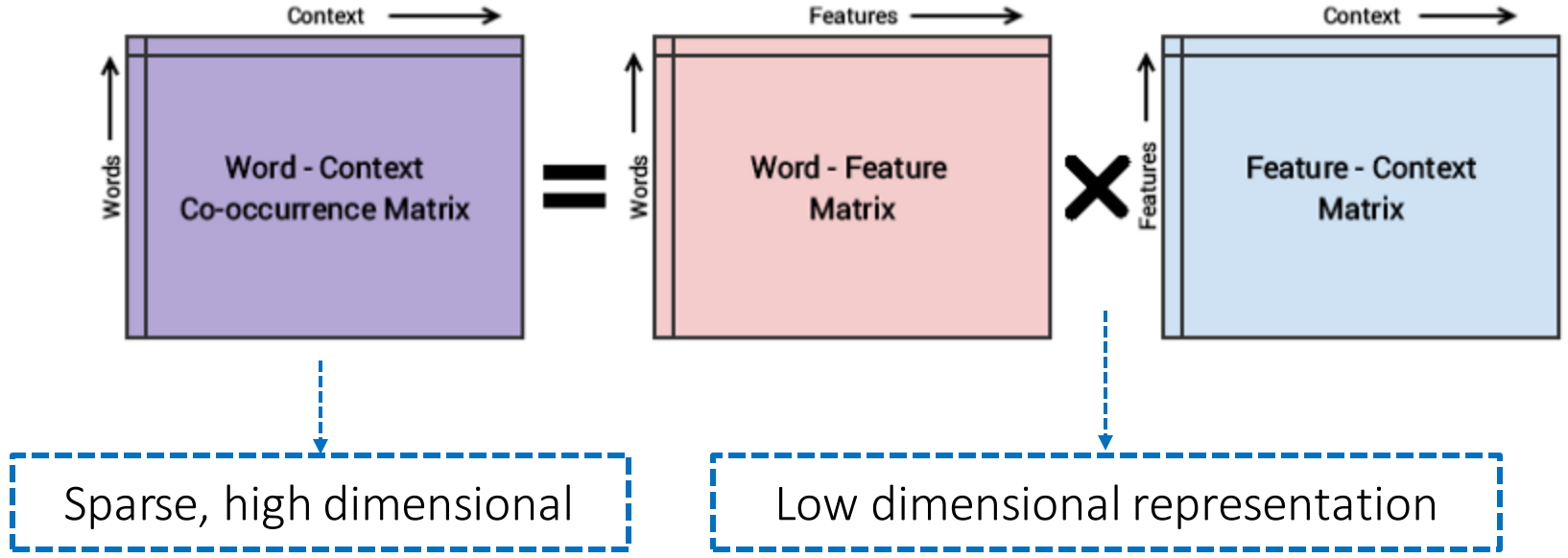
Research on Embedding Is Exploding Since Word2vec

- Word2vec is not a deep neural network, but it turns text into a numerical form that deep nets can understand
- *Neural word embedding* represents a word with a vector of numbers
  - Word2vec is similar to an autoencoder, encoding each word in a vector
    - It trains words against other words that neighbor them in the input corpus
- Applications: Analysis of text, genes, code, likes, other verbal/symbolic series
- Many powerful embedding methods generated since then
  - GloVe (Global Vectors for Word Representation) *J. Pennington, R. Socher, C. D. Manning (Stanford U.) EMNLP 2014*
  - fastText: represents each word as an n-gram of characters, allows the embeddings to understand suffixes and prefixes
  - Network embedding: Embedding nodes in information networks
    - LINE, DeepWalk, Metapath2Vec, ……

Unsupervised Word Embedding: GloVe

- GloVe factorizes a global co-occurrence matrix derived from the entire corpus
- Low-dimensional representations are obtained by solving a least-squares problem to "recover" the co-occurrence matrix

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$



Pennington, J., Socher, R., & Manning, C.D. (2014). Glove: Global Vectors for Word Representation. EMNLP.

Unsupervised Word Embedding: fastText

- fastText improves upon Word2Vec by incorporating subword information into word embedding

Tri-gram extraction

<where> ⟹ <wh, whe, her, ere, re>

- fastText allows sharing subword representations across words, since words are represented by the aggregation of their n-grams

Word2Vec probability expression

$$p(w_O|w_I) = \frac{\exp\left(v'_{w_O}{}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left(v'_{w}{}^{\top} v_{w_I}\right)}$$

$$\sum_{g \in \mathcal{G}_w} \mathbf{z}_g^{\top} \mathbf{v}_c.$$

Represent a word by the sum of the vector representations of its n-grams

N-gram embedding

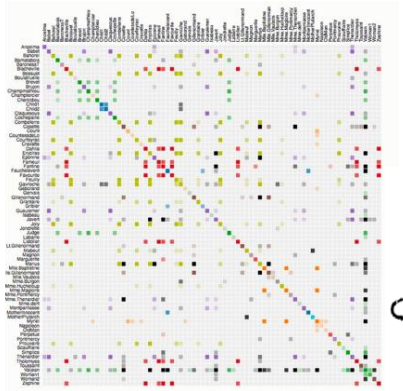
Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5, 135-146.

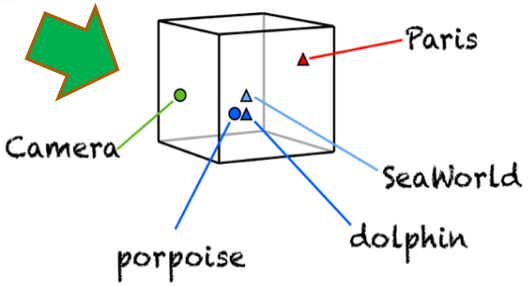
Outline

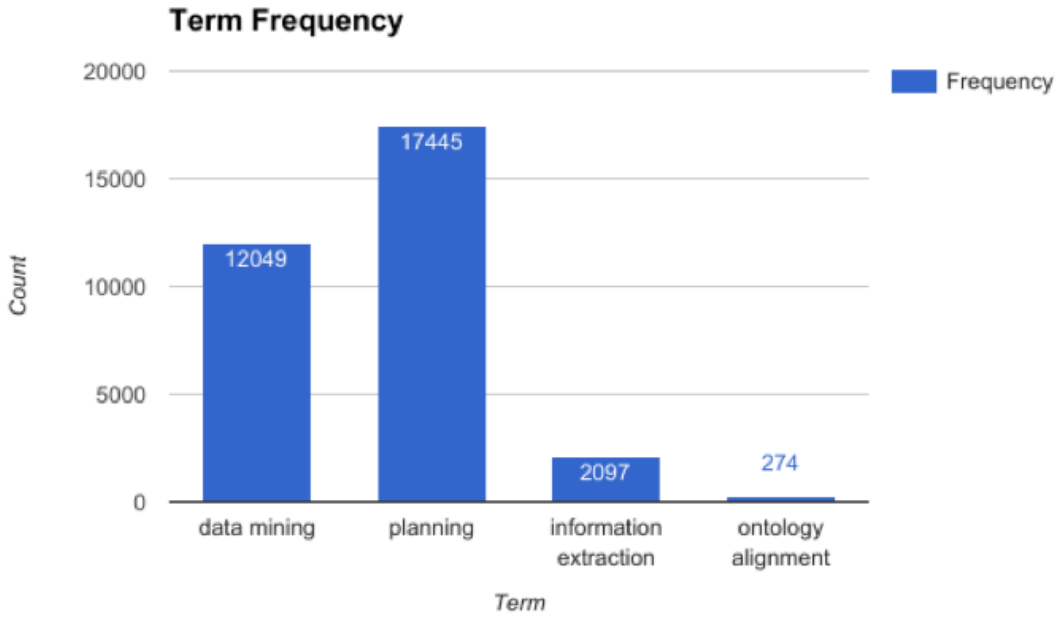
- Text Mining, Text Similarity and Text Embedding
- Unsupervised word embedding
  - Context-free representation:
    - Euclidean Embedding: Word2Vec, GloVe, fastText
      - Local-Corpus Based Embedding
    - Hyperbolic Embedding: Poincaré and Lorentz Embedding
    - Spherical Text Embedding: JoSE (Joint Spherical Text Embedding)
  - Contextualized representation: ELMo, BERT, XLNet
- CatE: Category-Name Guided Text Embedding for Topic Mining
- Looking Forward

Problem: Expert Finding in Bibliographic Networks

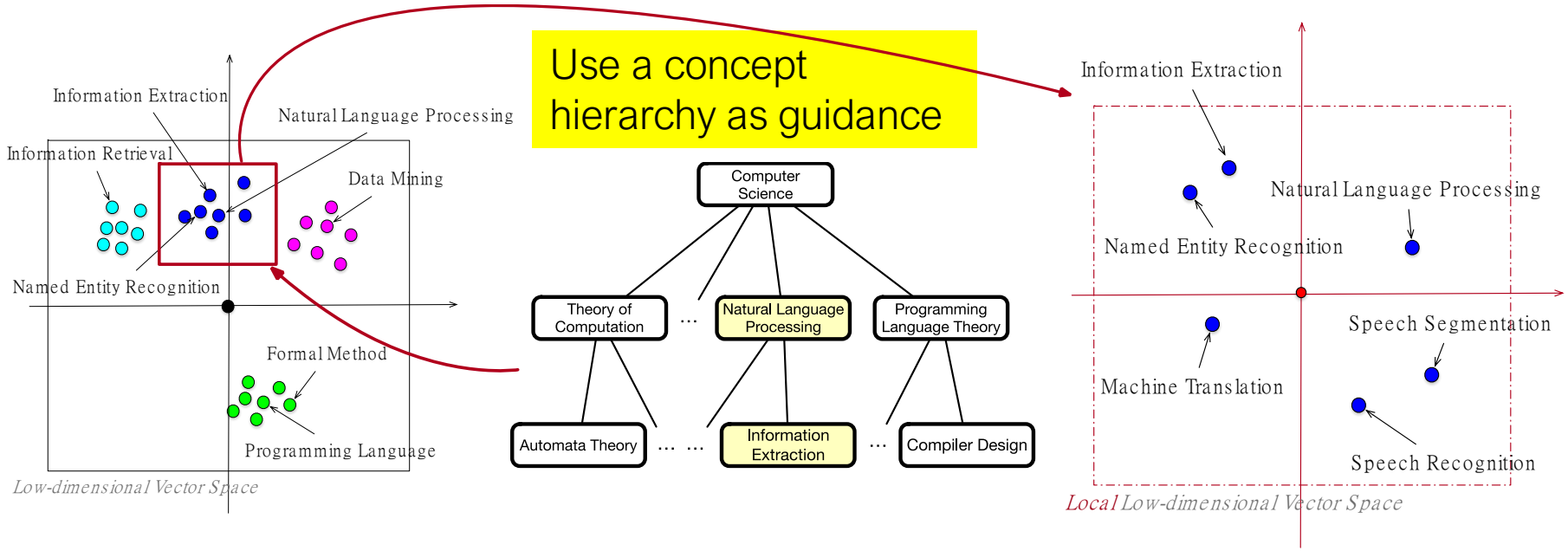
- Given a set of keywords, find related experts
  - Ex. Find expert on “information extraction”
- Challenges: Vocabulary gap
  - “*relation extraction*”, “*named entity recognition*”, …
- The power of word embedding
  - Use word embedding to close the vocabulary gap
- Difficulty: Discrepancy in queries
  - Specific queries: Narrow semantic meanings
    - “Information Extraction”
    - “Ontology Alignment”
  - General queries: Broad semantic meanings
    - “Data Mining”
    - “Planning”







Local Embedding Training with Concept Hierarchy



- For an arbitrary query, local embedding can be learned with the sub-corpus constrained on the parent topic — The parent topic becomes background
- Recursive Local Embedding Training
- The idea was proposed and developed by Huan Gui, et al. 2017 "Expert Finding in Heterogeneous Bibliographic Networks with Locally-trained Embeddings" (https://arxiv.org/pdf/1803.03370.pdf, deposited on March 2018)

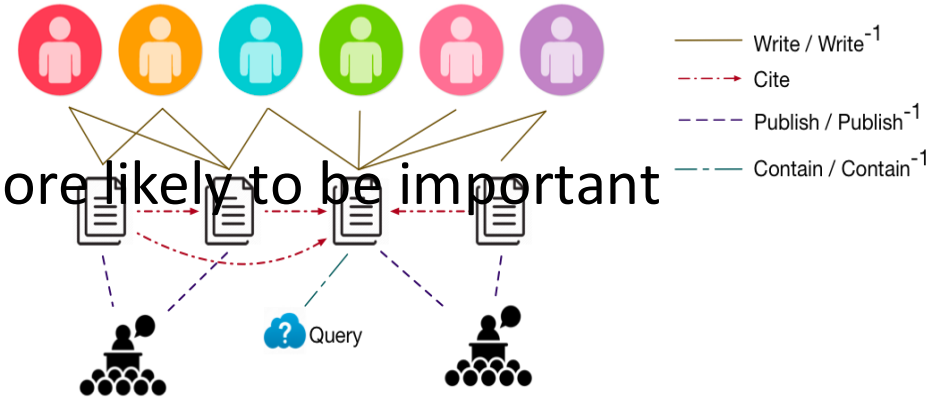
Ranking Experts in Heterogeneous Information Networks

- Expert Finding: Based on both **relevance** and **importance**
- Ranking in networks
  - Relevance Network
    - A candidate may have expertise on multiple topics
    - Only papers relevant to the query can serve as evidence
  - Heterogeneous Information Networks
    - Citation may have time-delay factor
    - Papers published in a higher-ranked venue are more likely to be important
    - Venues play an important role for ranking
- Ranking Philosophy
  - Important & relevant papers will be cited by many important & relevant papers
  - Relevant experts will publish many important & relevant papers
  - Relevant conferences will publish many important & relevant papers

Andrew McCallum

Information Extraction

Social Network analysis



Experiments: LE-expert vs. Other Methods

Dataset (DBLP):

Documents: 2,244,018

Authors: 1,274,360

Labels (20 queries):

General: machine-learning, natural-language-processing, planning

Specific: face-recognition, information-extraction, kernel-methods, ontology-alignment…

measure	P@5	P@10	P@20	NDCG@5	NDCG@10	NDCG@20	MAP	bpref
BALOG	0.4941	0.3824	0.2853	0.5068	0.4248	0.3416	0.1608	0.8536
NMF	0.3176	0.2706	0.2118	0.3525	0.3075	0.253	0.1151	0.7303
SVD	0.4353	0.3471	0.2912	0.4553	0.3871	0.3336	0.1548	0.7590
CORANK	0.6941	0.5741	0.4235	0.7181	0.6386	0.5024	0.291	0.8843
EMBED	0.0353	0.0294	0.0265	0.0354	0.0317	0.0289	0.005	0.6331
JOINTHYP	0.6235	0.4176	0.2882	0.6447	0.4913	0.3725	0.1579	**0.9704**
EXACT	0.7059	0.5882	0.4529	0.7548	0.6549	0.5361	0.311	0.8676
RankClass	0.7529	0.6647	0.5176	0.7666	0.7026	0.5867	0.3598	0.8981
LE-expert	**0.8118**	**0.7118**	**0.5559**	**0.8027**	**0.7361**	**0.618**	**0.3826**	0.9451
↑ vs BALOG	64.30%	86.14%	94.84%	58.38%	73.28%	80.91%	137.93%	10.73 %

Significant improvement compared with document-based model (BALOG)

Case Study

boosting		support vector machine	
Co-ranking	LE-expert	Co-ranking	LE-expert
Robert E. Schapire	**Robert E. Schapire**	Qi Wu	**Bernhard Scholkopf**
Yoav Freund	**Yoav Freund**	Isabelle Guyon	**Vladimir Vapnik**
Ron Kohavi	**Leo Breiman**	**Jason Weston**	**Christopher J. C. Burges**
Thomas G. Dietterich	**Yoram Singer**	**Vladimir Vapnik**	**Thorsten Joachims**
Yoram Singer	**David P. Helmbold**	Bao-Liang Lu	**Chih-Jen Lin**
information extraction		ontology alignment	
Co-ranking	LE-expert	Co-ranking	LE-expert
Ralph Grishman	**Dayne Freitag**	**Jerome euzenat**	**W. Marco Schorlemmer**
Andrew McCallum	**Ralph Grishman**	Patrick Lambrix	**Yannis Kalfoglou**
Ellen Riloff	**Andrew McCallum**	Jason J. Jung	**Anhai Doan**
Oren Etzioni	**Nicholas Kushmerick**	He Tan	**Jerome Euzenat**
Dayne Freitag	**Stephen Soderland**	Marc Ehrig	Alon Y. Halevy

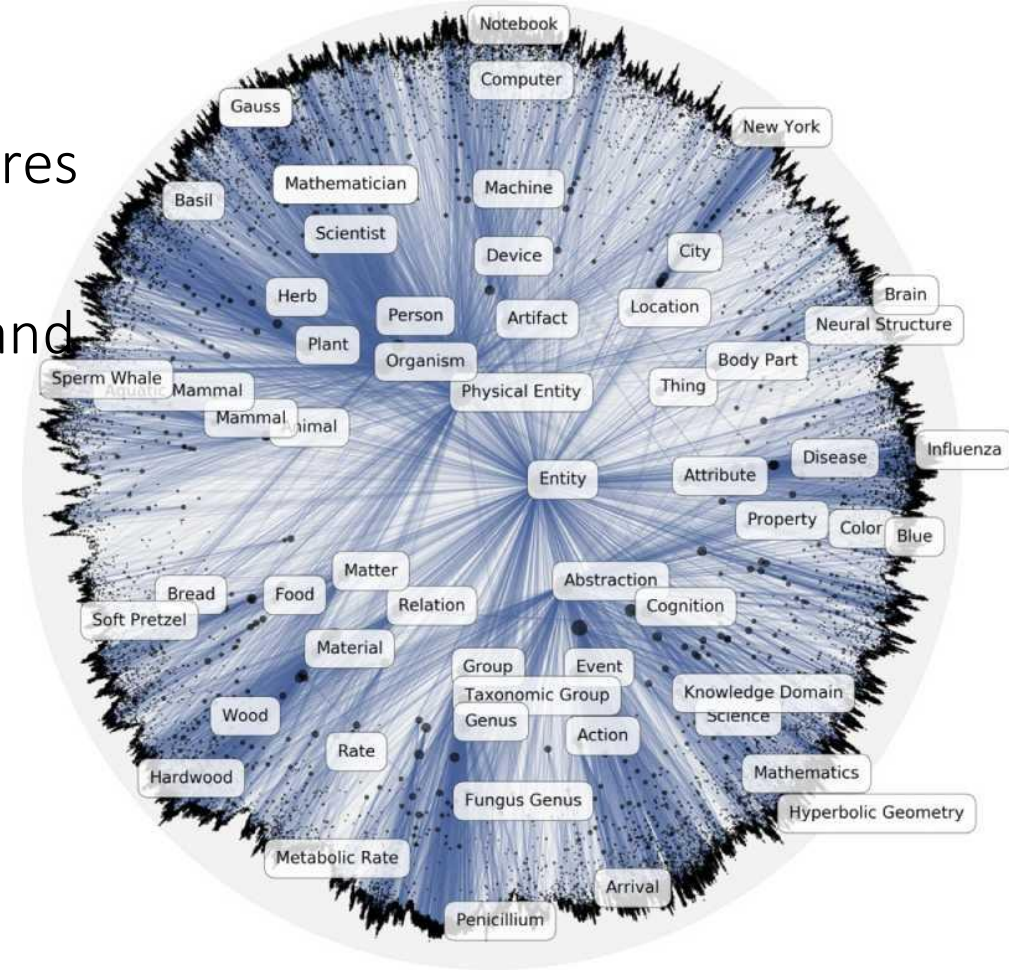
Outline

- Text Mining, Text Similarity and Text Embedding
- Unsupervised word embedding
  - Context-free representation:
    - Euclidean Embedding: Word2Vec, GloVe, fastText
      - Local-Corpus Based Embedding
    - Hyperbolic Embedding: Poincaré and Lorentz Embedding
    - Spherical Text Embedding: JoSE (Joint Spherical Text Embedding)
  - Contextualized representation: ELMo, BERT, XLNet
- CatE: Category-Name Guided Text Embedding for Topic Mining
- Looking Forward

Hyperbolic Embedding: Poincaré embedding

- Hyperbolic space = continuous version of trees
- Naturally equipped to model hierarchical structures
- Learns hierarchical representations by pushing general terms to the origin of the Poincaré ball, and specific terms to the boundary

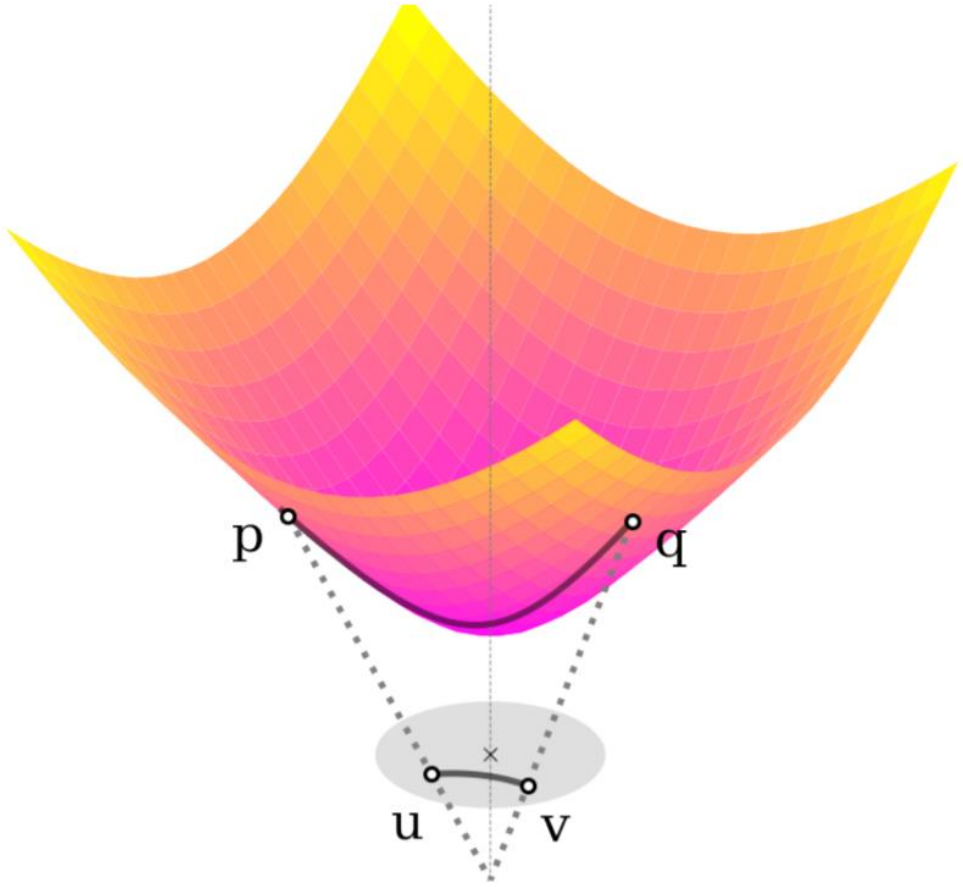
$$d(\boldsymbol{u}, \boldsymbol{v}) = \operatorname{arcosh}\left(1 + 2\frac{\|\boldsymbol{u} - \boldsymbol{v}\|^2}{(1 - \|\boldsymbol{u}\|^2)(1 - \|\boldsymbol{v}\|^2)}\right)$$



Nickel, M., & Kiela, D. (2017). Poincaré Embeddings for Learning Hierarchical Representations. NIPS.

Hyperbolic Embedding: Lorentz Embedding

- Extend the previous Poincaré embedding
- Comparison between Poincaré vs. Lorentz:
  - The Poincaré disk is intuitive for visualizing and interpreting hyperbolic embeddings
  - The Lorentz model is well-suited for Riemannian optimization
- Strategy:
  - Learn embedding via Lorentz model
  - Project embedding into the Poincaré disk



Nickel, M., & Kiela, D. (2018). Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. ICML.

Outline

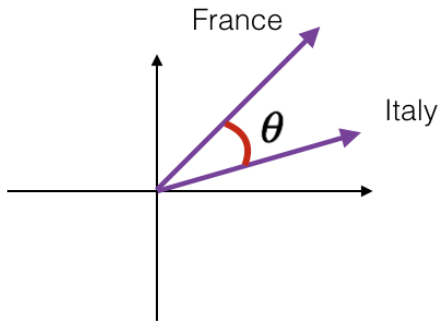
- Text Mining, Text Similarity and Text Embedding
- Unsupervised word embedding
  - Context-free representation:
    - Euclidean Embedding: Word2Vec, GloVe, fastText
      - Local-Corpus Based Embedding
    - Hyperbolic Embedding: Poincaré and Lorentz Embedding
    - Spherical Text Embedding: JoSE (Joint Spherical Text Embedding)
  - Contextualized representation: ELMo, BERT, XLNet
- CatE: Category-Name Guided Text Embedding for Topic Mining
- Looking Forward

Spherical Text Embedding

- Spherical Text Embedding:
  - "**Spherical**": Embeddings are trained on the unit sphere, where vector norms are ignored and directional similarity is directly optimized
  - "**Text Embedding**": Instead of training word embeddings only, we jointly train paragraph (document) embeddings with word embeddings to capture the local and global contexts in text embedding
- Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan and Jiawei Han, "Spherical Text Embedding", NeurIPS, Dec. 2019
  - A spherical generative model that jointly exploits word-word (local) and word-paragraph (global) co-occurrence statistics
  - An efficient optimization algorithm in the spherical space with convergence guarantee
  - State-of-the-art performance on various text embedding applications

Why Spherical Text Embedding?

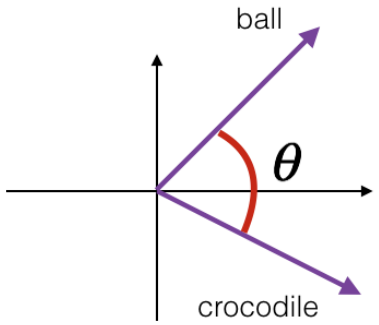
- Previous text embeddings (e.g., Word2Vec) are trained in the Euclidean space
  - But used on spherical space—Mostly directional similarity (i.e., cosine similarity)
  - Word similarity is derived using cosine similarity



France and Italy are quite similar

θ is close to 0°

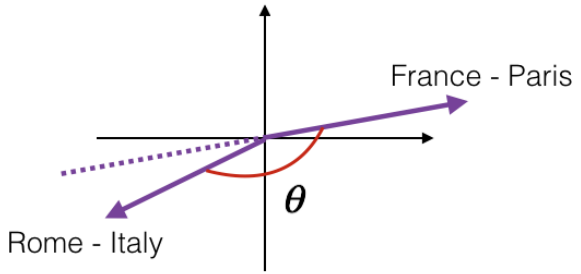
$\cos(\theta) \approx 1$



ball and crocodile are not similar

θ is close to 90°

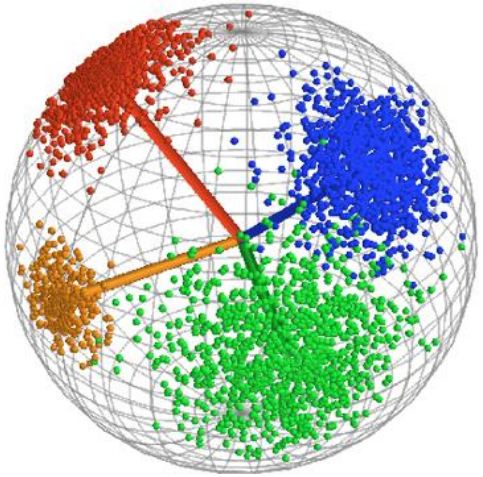
$\cos(\theta) \approx 0$



the two vectors are similar but opposite
the first one encodes (city - country)
while the second one encodes (country - city)

θ is close to 180°

$\cos(\theta) \approx -1$



  - Word clustering (e.g., TaxoGen) is performed on a sphere
  - Better document clustering performances when embeddings are normalized and spherical clustering algorithms are used

Inconsistency Between Training and Usage Space

- The inconsistency between word embedding training and usage space
  - The objective we optimize during training is not really the one we use
  - Regardless of the different training objective, Word2Vec, GloVe and fastText all optimize the embedding **dot product** during training, but **cosine similarity** is what actually used in applications

Embedding dot product is optimized during training

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^{\top} v_{w_I}\right)}$$

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

$$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^{\top} \mathbf{v}_c$$

Word2Vec GloVe fastText

Consequence of Such Inconsistency

- Consequence: The objective we optimize during training is not really the one we use
  - Ex: Consider two pairs of words, A: lover-quarrel; B: rock-jazz
  - Pair B has higher ground truth similarity than pair A in WordSim353 (a benchmark test set)
  - Word2Vec assigns higher dot product to pair B, but its cosine similarity is still smaller than pair A

Metrics	A: *lover-quarrel*		B: *rock-jazz*
Dot Product	5.284	<	6.287
Cosine Similarity	0.637	>	0.628

Training

Usage

Inconsistency

Another Observation: Integrating Local and Global Contexts

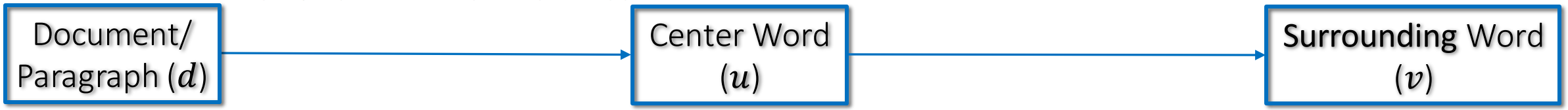
- Local contexts can only partly define word semantics in unsupervised word embedding learning

Local contexts of "harmful"

If I hear someone screwing with my car (ie, setting off the **alarm**) and **taunting** me to come out, you can be very sure that my Colt Delta Elite will also be coming with me. It is not the screwing with the car that would get them **shot**, it is the potential physical **danger**. If they are **taunting** like that, it's very possible that they also intend to **rob** me and or do other physically *harmful* things. Here in Houston last year a woman heard the sound of someone …

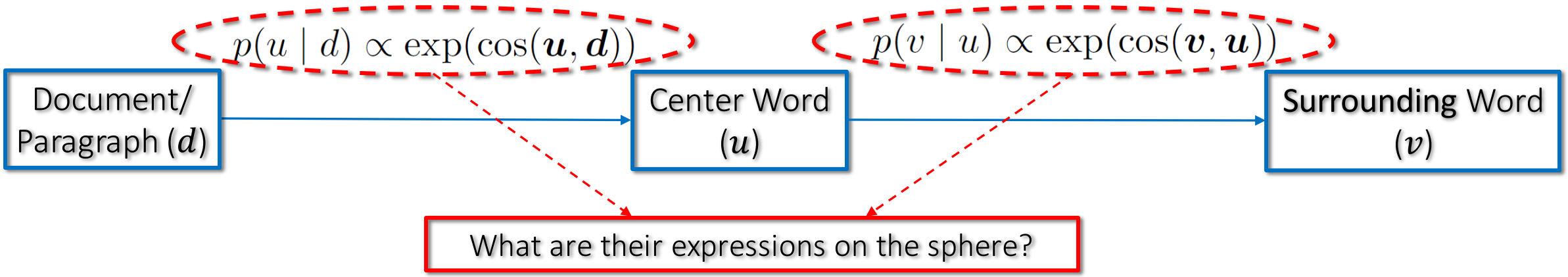
- Design a generative model on the sphere that follows how humans write articles:
  - First a general idea of the paragraph/doc, then start to write down each word in consistent with not only the paragraph/doc, but also the surrounding words

$p(u \mid d) \propto \exp(\cos(\boldsymbol{u}, \boldsymbol{d}))$ $p(v \mid u) \propto \exp(\cos(\boldsymbol{v}, \boldsymbol{u}))$



Model: Spherical Text Embedding

- How to define the generative model in the spherical space?

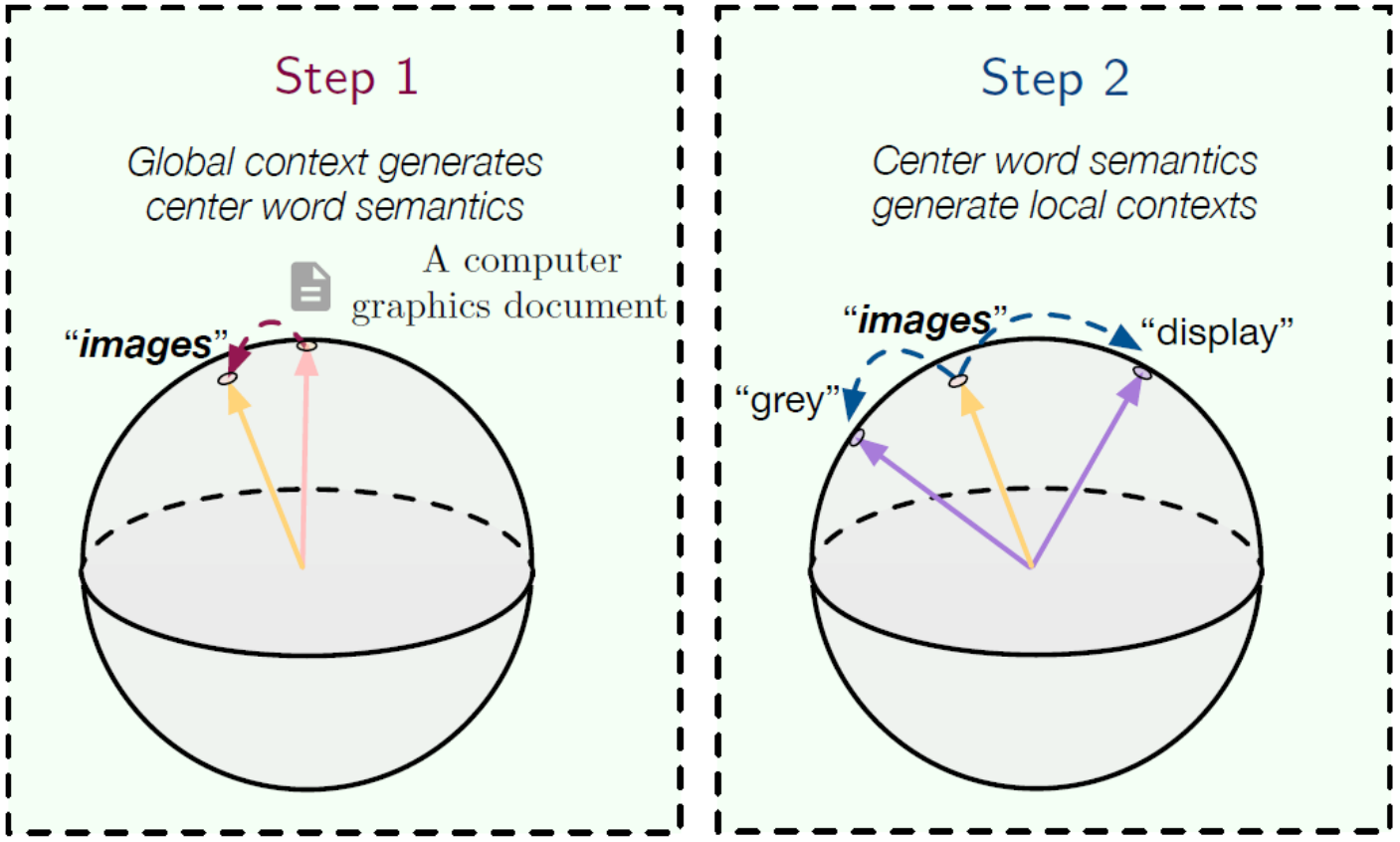


- A theorem connecting the above generative model with a spherical probability distribution

Theorem 1. When the corpus has infinite vocabulary, *i.e.*, $|V| \to \infty$, the analytic forms of $p(u \mid d) \propto \exp(\cos(\boldsymbol{u}, \boldsymbol{d}))$ and $p(v \mid u) \propto \exp(\cos(\boldsymbol{v}, \boldsymbol{u}))$ are given by the von Mises-Fisher (vMF) distribution with the prior embedding as the mean direction and constant 1 as the concentration parameter, *i.e.*,

$$\lim_{|V|\to\infty} p(v \mid u) = \text{vMF}_p(\boldsymbol{v}; \boldsymbol{u}, 1), \qquad \lim_{|V|\to\infty} p(u \mid d) = \text{vMF}_p(\boldsymbol{u}; \boldsymbol{d}, 1).$$

Understanding the Spherical Generative Model



Modeling Spherical Text Embedding

- Training objective:
  - The final generation probability:

$$p(v, u \mid d) = p(v \mid u) \cdot p(u \mid d) = \text{vMF}_p(\boldsymbol{v}; \boldsymbol{u}, 1) \cdot \text{vMF}_p(\boldsymbol{u}; \boldsymbol{d}, 1)$$

  - Maximize the log-probability of a real co-occurred tuple (v, u, d), while minimize that of a negative sample (v, u', d), with a max-margin loss:

$$\mathcal{L}_{\text{joint}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{d}) = \max\Big(0, m - \log\left(c_p(1)\exp(\cos(\boldsymbol{v}, \boldsymbol{u})) \cdot c_p(1)\exp(\cos(\boldsymbol{u}, \boldsymbol{d}))\right) + \log\left(c_p(1)\exp(\cos(\boldsymbol{v}, \boldsymbol{u}')) \cdot c_p(1)\exp(\cos(\boldsymbol{u}', \boldsymbol{d}))\right)\Big)$$

Positive Sample

Negative Sample

$$= \max\left(0, m - \cos(\boldsymbol{v}, \boldsymbol{u}) - \cos(\boldsymbol{u}, \boldsymbol{d}) + \cos(\boldsymbol{v}, \boldsymbol{u}') + \cos(\boldsymbol{u}', \boldsymbol{d})\right),$$

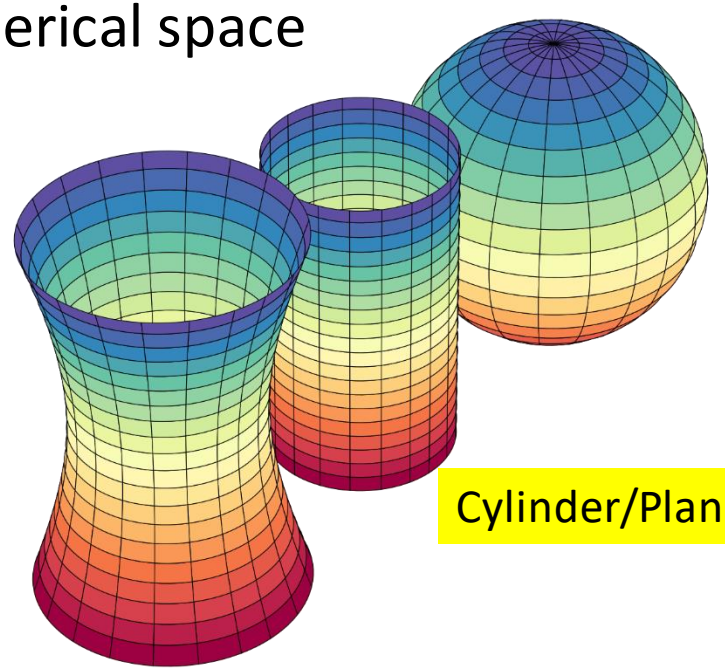
Computation: Optimization on the Sphere

- The constrained optimization problem:

$$\min_{\Theta} \mathcal{L}_{\text{joint}}(\Theta) \quad \text{s.t.} \quad \forall \boldsymbol{\theta} \in \Theta : \|\boldsymbol{\theta}\| = 1 \qquad \Theta = \{\boldsymbol{u}_i\}\big|_{i=1}^{|V|} \cup \{\boldsymbol{v}_i\}\big|_{i=1}^{|V|} \cup \{\boldsymbol{d}_i\}\big|_{i=1}^{|\mathcal{D}|}$$

- Challenge: Parameters must be always updated on the sphere, but Euclidean optimization methods (e.g., SGD) are not constrained on a curvature space
- Need to consider the nature of the spherical space
  - Riemannian manifold:

The sphere is a Riemannian manifold with constant positive curvature



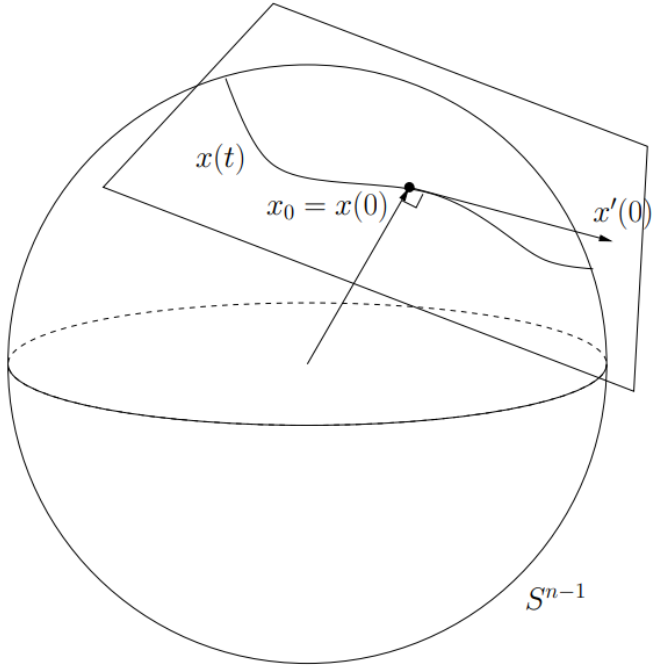
Riemannian Optimization with Riemannian SGD

- Riemannian gradient:

$$\operatorname{grad} f(\boldsymbol{x}) := \left(I - \boldsymbol{x}\boldsymbol{x}^\top\right) \nabla f(\boldsymbol{x})$$

- Exponential mapping (maps from the tangent plane to the sphere):

$$\exp_{\boldsymbol{x}}(\boldsymbol{z}) := \begin{cases} \cos(\|\boldsymbol{z}\|)\boldsymbol{x} + \sin(\|\boldsymbol{z}\|)\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}, & \boldsymbol{z} \in T_{\boldsymbol{x}}\mathbb{S}^{p-1} \backslash \{\mathbf{0}\}, \\ \boldsymbol{x}, & \boldsymbol{z} = \mathbf{0}. \end{cases}$$



- Riemannian SGD:

$$\boldsymbol{x}_{t+1} = \exp_{\boldsymbol{x}_t}\left(-\eta_t \operatorname{grad} f(\boldsymbol{x}_t)\right)$$

- Retraction (first-order approximation of the exponential mapping):

$$R_{\boldsymbol{x}}(\boldsymbol{z}) := \frac{\boldsymbol{x} + \boldsymbol{z}}{\|\boldsymbol{x} + \boldsymbol{z}\|}$$

Word Similarity: Experiment Setting

- Data and Measures
  - Train 100-d word embedding on the latest Wikipedia dump (~13G)
  - Compute embedding **cosine similarity** between word pairs to obtain a ranking of similarity
  - Benchmark datasets contain human rated similarity scores
  - The more similar the two rankings are, the better embedding reflects human thoughts
  - **Spearman's rank correlation** is used to measure the ranking similarity
- Baselines
  - Euclidian Space: Word2Vec, GloVe, fastText, BERT (NAACL, 2019)
  - Poincare Space: Poincaré glove: Hyperbolic word embeddings (ICLR, 2019)
  - Spherical Space: JoSE (our full model)

Word Similarity: Performance Comparison

Table 1: Spearman rank correlation on word similarity evaluation.

Embedding Space	Model	WordSim353	MEN	SimLex999
Euclidean	Word2Vec	0.711	0.726	0.311
	GloVe	0.598	0.690	0.321
	fastText	0.697	0.722	0.303
	BERT	0.477	0.594	0.287
Poincaré	Poincaré GloVe	0.623	0.652	0.321
Spherical	**JoSE**	**0.739**	**0.748**	**0.339**

- Why does BERT fall behind on this task?
  - BERT learns contextualized representations, but word similarity is conducted in a context-free manner
  - BERT is optimized on specific downstream tasks like predicting masked words and sentence relationships, which have no direct relation to word similarity

Document Clustering: Experiment Setting

- Data and Measures
  - Train document embedding on 20News dataset (20 classes)
  - Perform K-means and Spherical K-means (SK-means)
  - Metrics: Mutual Information (MI), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Purity
  - Run clustering 10 times and report the above metrics with mean and standard deviation
- Baselines:
  - Euclidian Space: Avg. Word2Vec, BERT
    - SIF: Simple but tough-to-beat baseline for sentence embeddings. In ICLR, 2017
    - Doc2Vec: Distributed representations of sentences and documents. In ICML, 2014
  - Spherical Space: JoSE (our full model)

Document Clustering: Performance Comparison

Table 2: Document clustering evaluation on the 20 Newsgroup dataset.

Embedding	Clus. Alg.	MI	NMI	ARI	Purity
Avg. W2V	K-Means	1.299 ± 0.031	0.445 ± 0.009	0.247 ± 0.008	0.408 ± 0.014
	SK-Means	1.328 ± 0.024	0.453 ± 0.009	0.250 ± 0.008	0.419 ± 0.012
SIF	K-Means	0.893 ± 0.028	0.308 ± 0.009	0.137 ± 0.006	0.285 ± 0.011
	SK-Means	0.958 ± 0.012	0.322 ± 0.004	0.164 ± 0.004	0.331 ± 0.005
BERT	K-Means	0.719 ± 0.013	0.248 ± 0.004	0.100 ± 0.003	0.233 ± 0.005
	SK-Means	0.854 ± 0.022	0.289 ± 0.008	0.127 ± 0.003	0.281 ± 0.010
Doc2Vec	K-Means	1.856 ± 0.020	0.626 ± 0.006	0.469 ± 0.015	0.640 ± 0.016
	SK-Means	1.876 ± 0.020	0.630 ± 0.007	0.494 ± 0.012	0.648 ± 0.017
JoSE	K-Means	1.975 ± 0.026	0.663 ± 0.008	0.556 ± 0.018	0.711 ± 0.020
	SK-Means	$\mathbf{1.982} \pm 0.034$	$\mathbf{0.664} \pm 0.010$	$\mathbf{0.568} \pm 0.020$	$\mathbf{0.721} \pm 0.029$

- Embedding quality is generally more important than clustering algorithms
  - Using spherical K-Means only gives marginal performance boost over K-Means
  - JoSE embedding remains optimal regardless of clustering algorithms

Document Classification: Experiment Results

- Train doc embedding on 20News (20 classes) and Movie review (2 classes) dataset
- Perform k-NN classification (k=3): similar comparison results with k from [1, 10]
  - k-NN is non-parametric and directly reflect how well the topology of the embedding space captures document-level semantics
- Metrics: Macro-F1 & Micro-F1
- Baselines: Same with document clustering

Table 3: Document classification evaluation using k-NN ($k = 3$).

Embedding	20 Newsgroup		Movie Review	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Avg. W2V	0.630	0.631	0.712	0.713
SIF	0.552	0.549	0.650	0.656
BERT	0.380	0.371	0.664	0.665
Doc2Vec	0.648	0.645	0.674	0.678
JoSE	**0.703**	**0.707**	**0.764**	**0.765**

Training Efficiency & Case Studies

- Training efficiency
  - Other models' objectives contain many non-linear operations, whereas JoSE only has linear terms in the objective

Table 4: Training time (per iteration) on the latest Wikipedia dump.

Word2Vec	GloVe	fastText	BERT	Poincaré GloVe	**JoSE**
0.81 hrs	0.85 hrs	2.11 hrs	> 5 days	1.25 hrs	**0.73 hrs**

- Acronym → similar words

Table 5: Effect of Global Context on Interpreting Acronyms.

Acronyms	Global ($\lambda = \infty$)	Local ($\lambda = 0$)
CMU	**mellon**, **carnegie**, andrew, pa, pittsburgh	andrew, kfnjyea00uh, am2x, mr47, devineni
UIUC	**urbana**, **illinois**, uxa, **univ**, uchicago	uxa, ux4, ux1, mrcnext, cka52397
UNC	**chapel**, **carolina**, astro, images, usc	launchpad, gibbs, umr, lambada, jge
Caltech	**california**, gap, **institute**, keith, **technology**	juliet, jafoust, lmh, henling, bdunn
JHU	**johns**, camp, **hopkins**, nation, grand	pablo, hasch, iglesias, davidk, atlantis

- Testing antonym similarity

Table 6: Cosine Similarity of Antonym Embeddings Trained with Different Contexts.

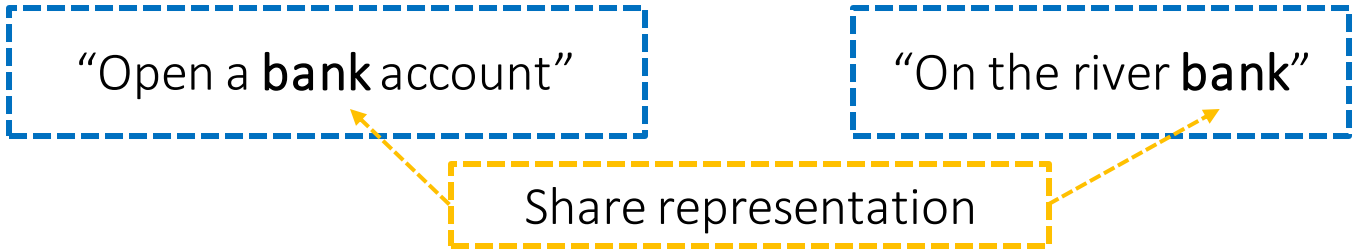
Antonyms	Global ($\lambda = \infty$)	Local ($\lambda = 0$)
good - bad	0.3150	0.7127
happy - unhappy	0.3911	0.6178
large - small	0.4871	0.7265
increase - decrease	0.2663	0.7308
enter - exit	0.2756	0.5553
save - spend	-0.0388	0.4792

Outline

- Text Mining, Text Similarity and Text Embedding
- Unsupervised word embedding
  - Context-free representation:
    - Euclidean Embedding: Word2Vec, GloVe, fastText
      - Local-Corpus Based Embedding
    - Hyperbolic Embedding: Poincaré and Lorentz Embedding
    - Spherical Text Embedding: JoSE (Joint Spherical Text Embedding)
  - Contextualized representation: ELMo, BERT, XLNet
- CatE: Category-Name Guided Text Embedding for Topic Mining
- Looking Forward

From Context-Free Embedding to Contextual Embedding

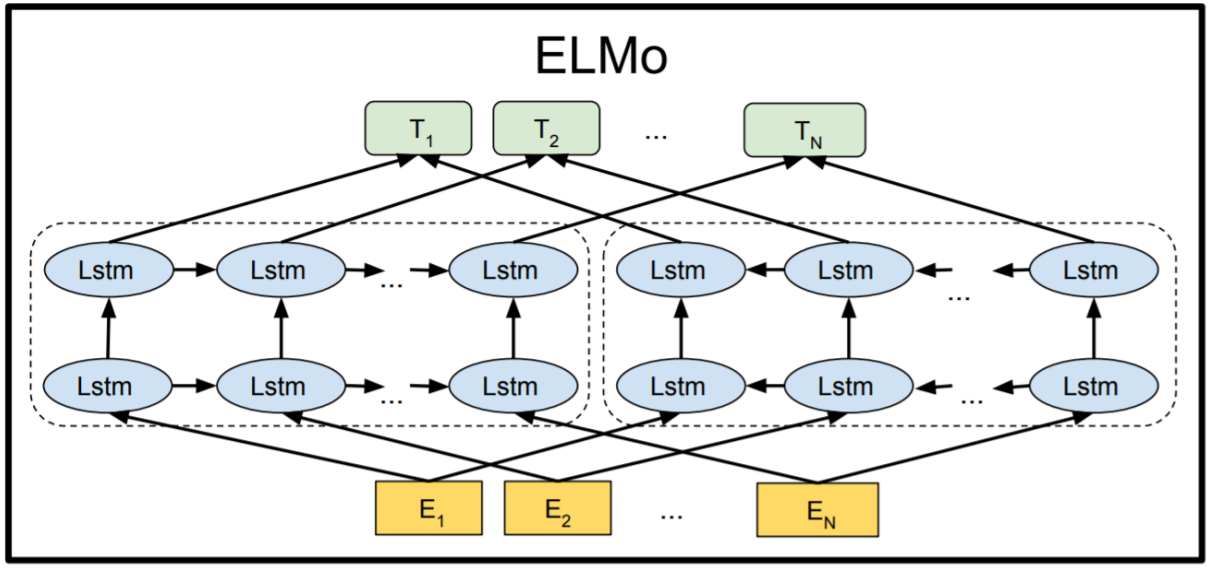
- Previous unsupervised word embeddings like Word2Vec and GloVe learn **context-free** word embedding
  - Each word has one representation regardless of specific contexts it appears in
  - E.g., “bank” is a polysemy, but only has one representation



- Recent popular deep language models overcome this problem by learning **contextual** relations between words and sentences

ELMo: Deep Contextualized Word Representations

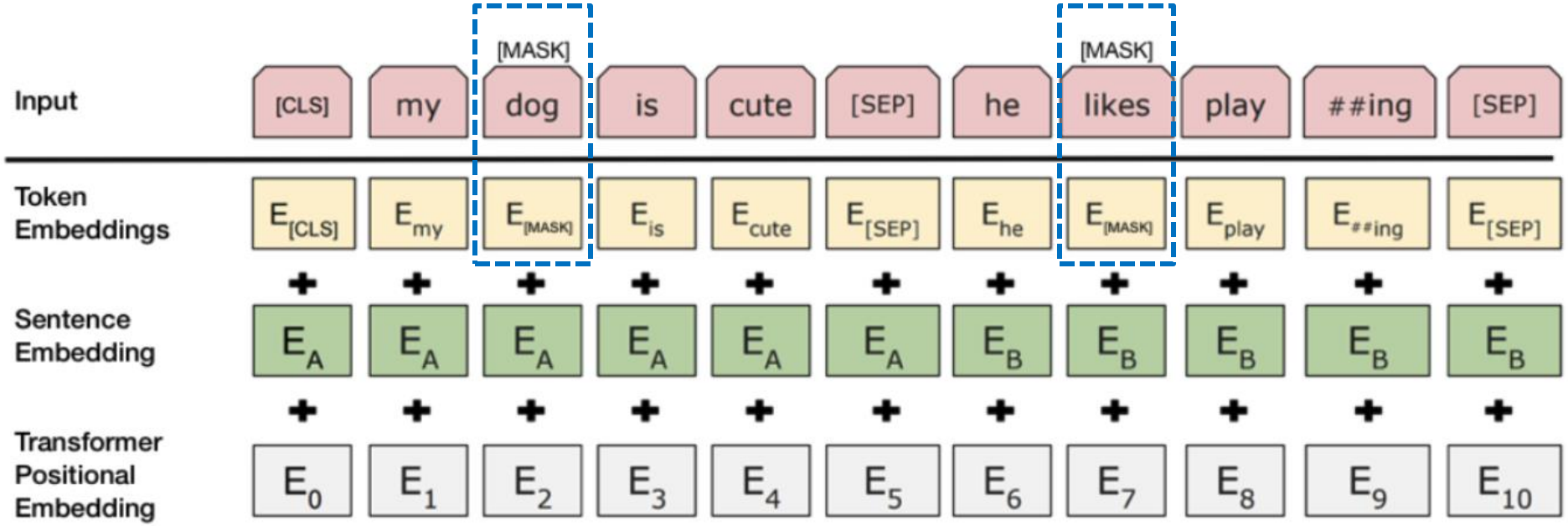
- Word representations are learned functions of the internal states of a deep bi-directional LSTMs
- Results in a pre-trained network that benefits several downstream tasks (e.g. Sentiment analysis, Named entity extraction, Question answering)
- However, left-to-right and right-to-left LSTMs are **independently** trained and concatenated



Peters, M.E., Neumann, M., Iyyer, M., Gardner, M.P., Clark, C., Lee, K., & Zettlemoyer, L.S. (2018). Deep contextualized word representations. NAACL.

BERT: Deep Bidirectional Transformers

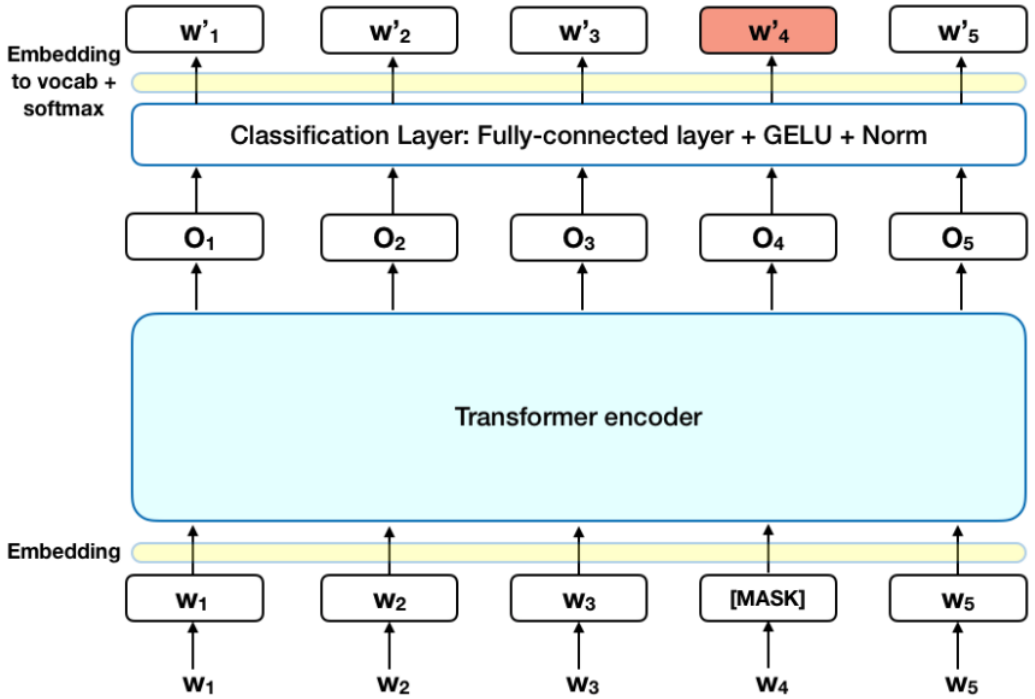
- Bidirectional: BERT leverages a Masked LM learning to introduce **real bidirectionality** training
- Masked LM: With 15% words randomly masked, the model learns bidirectional contextual information to predict the masked words



Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *NAACL* (2019).

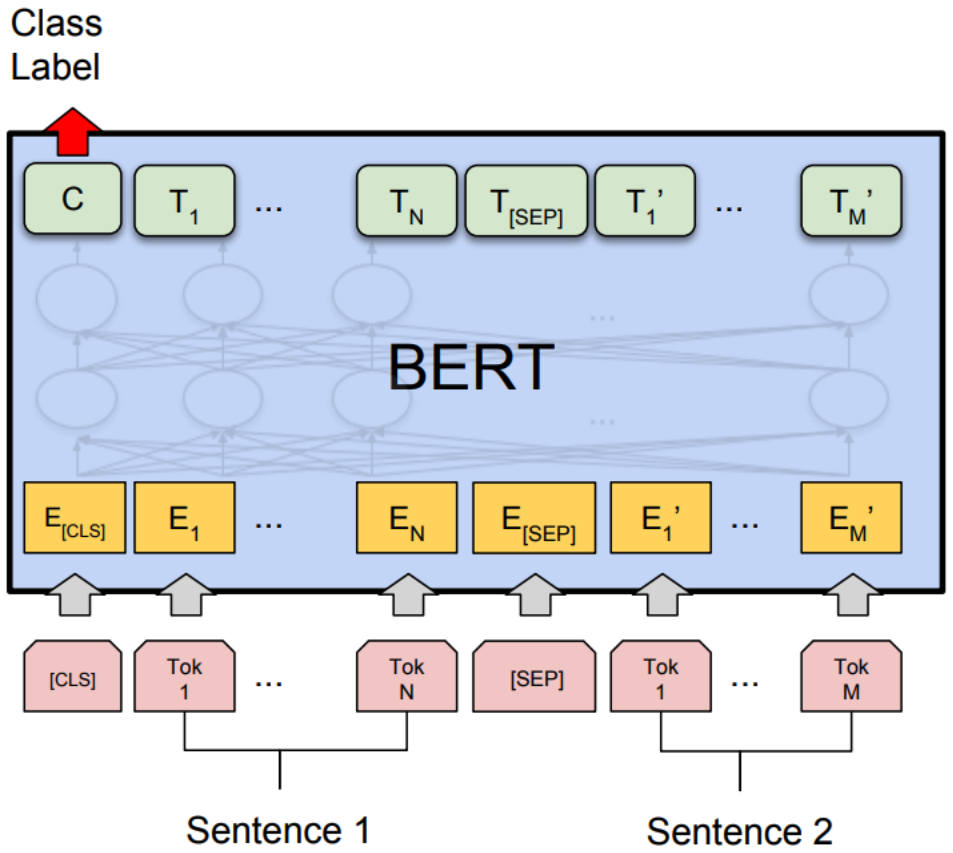
BERT: Deep Bidirectional Transformers

- Transformer Encoder: Reads the entire sequence of words at once; learns the context of a word based on all of its surroundings
- The Transformer employs an attention mechanism that learns contextual relations between words (and sub-words) in a text sequence



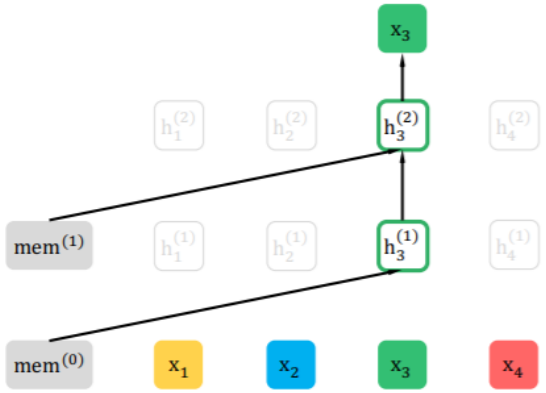
BERT: Deep Bidirectional Transformers

- Next Sentence Prediction: BERT receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document

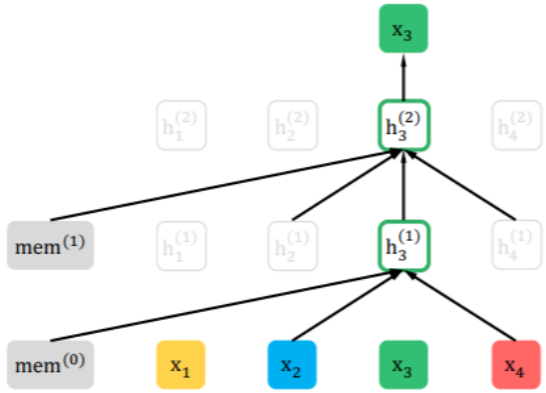


XLNet: Autoregressive Language Modeling

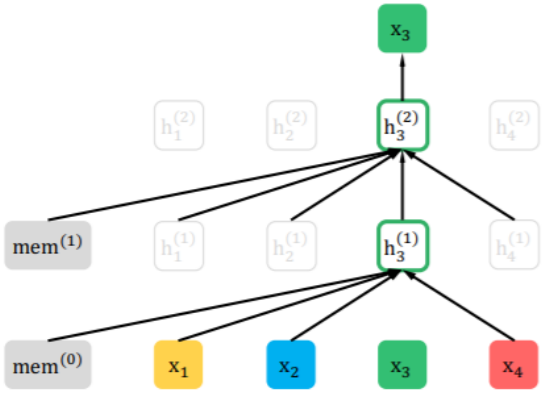
- Instead of using Masked LM, XLNet uses Permutation Language Modeling



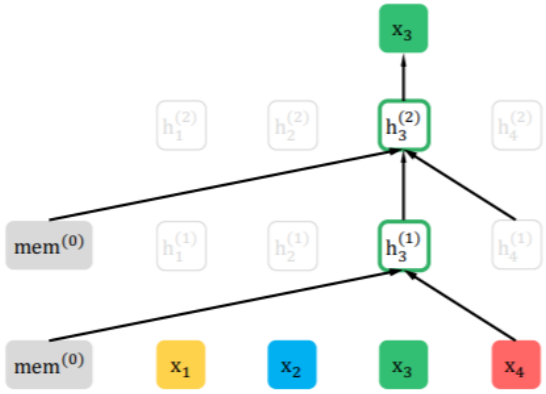
Factorization order: 3 → 2 → 4 → 1



Factorization order: 2 → 4 → 3 → 1



Factorization order: 1 → 4 → 2 → 3



Factorization order: 4 → 3 → 1 → 2

- Permutes the text sequence and predicts the target word using the remaining words in the sequence
- Since words in the original sequence are permuted, both forward direction information and backward direction information are leveraged

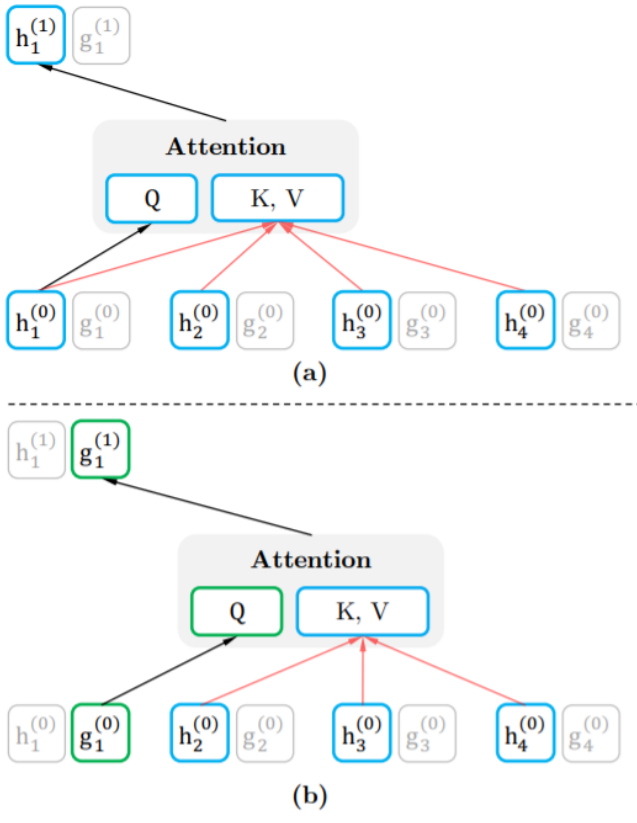
Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. NeurIPS'19

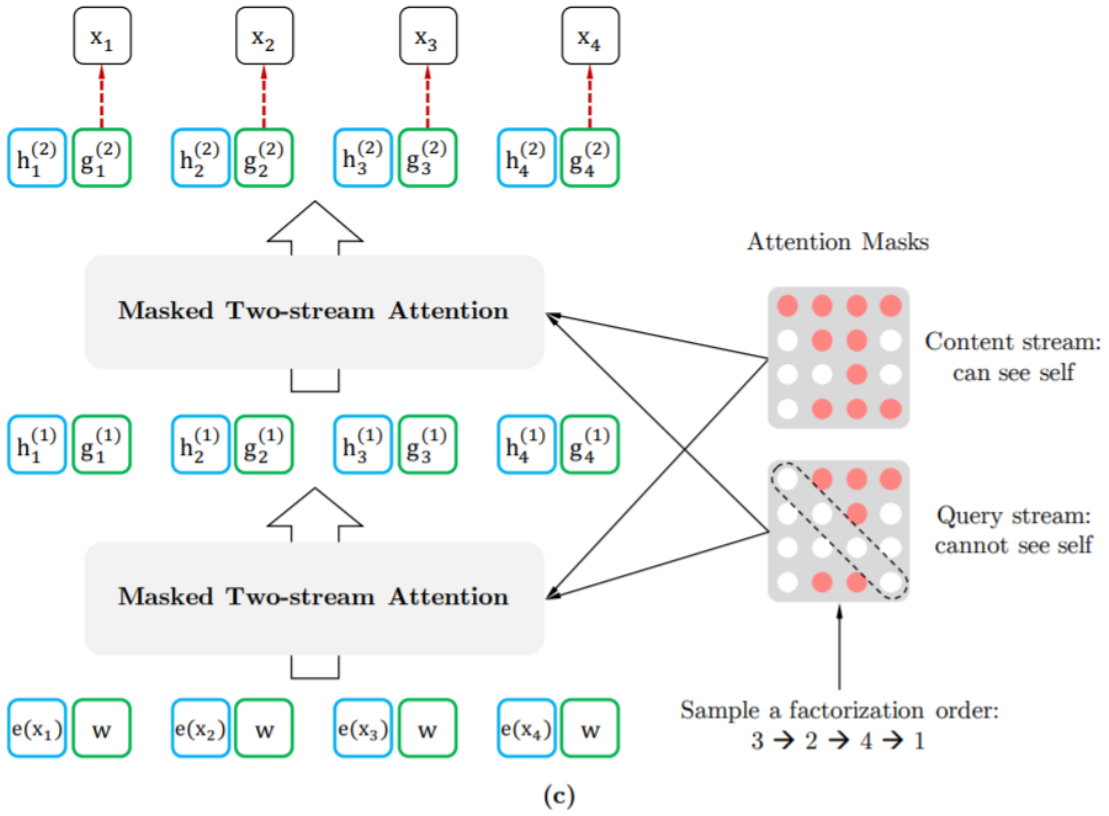
XLNet: Two-Stream Self-Attention

- Content representation: Encodes both token position as well as content
- Query representation: Encodes only token position

Content representation

Query representation





Outline

- Text Mining, Text Similarity and Text Embedding
- Unsupervised word embedding
  - Context-free representation:
    - Euclidean Embedding: Word2Vec, GloVe, fastText
      - Local-Corpus Based Embedding
    - Hyperbolic Embedding: Poincaré and Lorentz Embedding
    - Spherical Text Embedding: JoSE (Joint Spherical Text Embedding)
  - Contextualized representation: ELMo, BERT, XLNet
- CatE: Category-Name Guided Text Embedding for Topic Mining
- Looking Forward

From Unsupervised Embedding to Weakly-Supervised Embedding

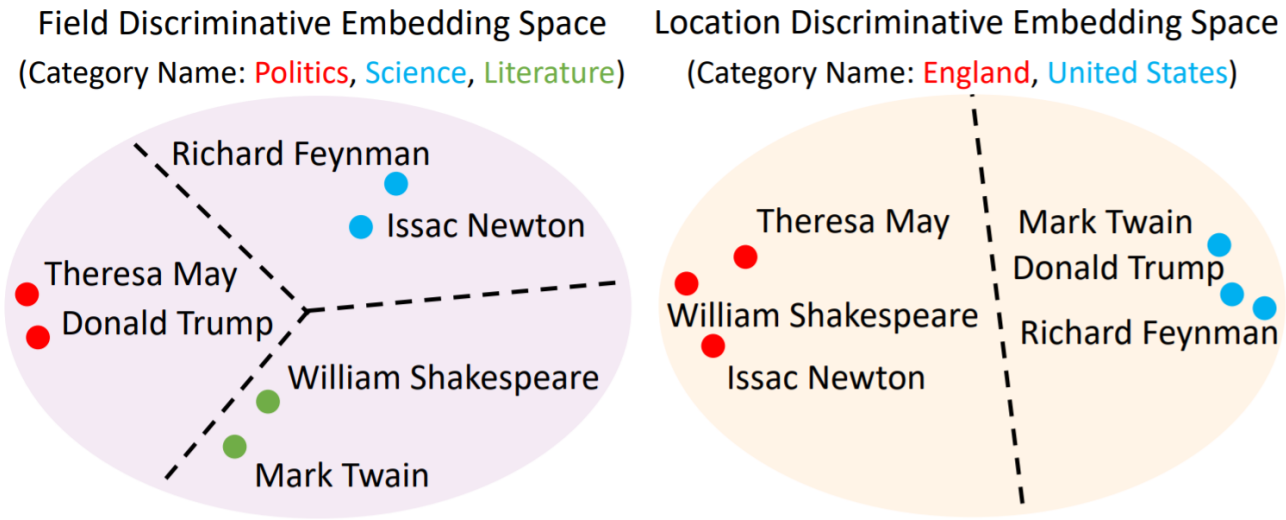
- Unsupervised word embedding can be used as word representations/features in a wide spectrum of text mining tasks
- However, unsupervised word embeddings are **generic** word representations
  - Not yielding the best performance on downstream tasks (e.g., taxonomy construction, document classification)
  - Reason: Not incorporating **task-specific** information
- We introduce a weakly-supervised embedding learning method called **CatE** that learns category distinctive embeddings
- Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang and Jiawei Han, "Discriminative Topic Mining via Category-Name Guided Text Embedding", in Proc. 2020 Int. World Wide Web Conf. (WWW'20), Taipei, Taiwan, Apr. 2020

Introduction: Idea of CatE

- **CatE** consists of two key modules:
  - A **weakly-supervised word embedding learning** module that regularizes the embedding space based on category representative words
  - A **selection** module that selects category representative words based on both word embedding similarity and word distributional specificity
- The two modules collaborate in an **iterative** way: At each iteration,
  - The former refines embeddings for accurate representative words selection
  - The latter selects representative words that will be used by the former in the next iteration

Introduction: Why Category Name Guided Embedding?

- **CatE** does not require any hand-labeled data, but only leverages **category names** to learn word embeddings with discriminative power over the specific set of categories
- Inputs: Category names + Corpus
  - E.g., the same set of celebrities should be embedded differently given different sets of category names



Topic Modeling and Its Challenges

- Topic models [LDA: Blei, Ng, Jordan 2003; pLSA: Hofmann 1999]
  - Unsupervised statistic tools that discover latent topics from text corpora
  - Optimized under a generative process: Maximal likelihood of the observed data
  - Tend to discover the most dominant or superficial topics: may not be of user's particular interest, or provide a skewed and biased summarization of the corpus
  - Not explicitly enforce the retrieved topics to be distinctive from each other
    - Example:

Table 1: LDA retrieved topics on NYT dataset. The meanings of the retrieved topics have overlap with each other.

Topic 1	Topic 2	Topic 3
canada, united states canadian, economy	sports, united states olympic, games	united states, iraq government, president

- Later developments
  - Supervised LDA and DiscLDA: But rely on massive hand-labeled documents
  - Seed-guided topic modeling: Still not impose requirements on the distinctiveness of the retrieved topics

Discriminative Topic Mining: General Philosophy

- Discriminative Topic Mining [Meng et al., WWW 2020]
  - Takes a set of category names as user guidance
  - Retrieve a set of representative and discriminative terms in each category
- Method: CatE: category-name guided text embedding for discriminative topic mining
  - A category-name guided text embedding learning module (E):
    - Takes a set of category names to learn category distinctive word embeddings by modeling the text generative process conditioned on the user provided categories
  - A category representative words retrieval module (R):
    - Selects category representative words based on both word embedding similarity and word distributional specificity
  - The two modules (E + R) collaborate in an iterative way:
    - E refines word embeddings and category embeddings
    - R selects representative words that will be used by E in the next iteration

Discriminative Topic Mining: Problem Formulation

- Definition: Given a text corpus D and a set of category names C = {c_1, . . . , c_n }
  - Discriminative topic mining aims to retrieve a set of terms S_i = {w_1, . . . , w_m} from D for each category c_i such that each term in S_i semantically belongs to and only belongs to category c_i
- Example: Given c_1: "The United States", c_2: "France" and c_3: "Canada"
  - Correct: "Ontario" in S_3 (a province in Canada; but not in other categories)
  - Incorrect: "North America" in S_3 (not belong to any countries)
  - Incorrect: "English" in S_3 (English is also the national language of the United States)
- Key points:
  - Requires a set of user-provided category names and only focuses on retrieving terms belonging to this specific set of categories
  - Imposes strong discriminative requirements: Each retrieved term must belong to and only belong to that category semantically

Related Work: Task-Oriented Text Embedding

- Supervised word embedding: Explicitly leverage category info. to optimize embedding; but require labeled training documents to fine-tune word embeddings
  - Predictive Text Embedding (PTE) [Tang et al,. KDD 2015]: constructs a heterogeneous text network and jointly embeds words, documents and labels based on word-word and word-document co-occurrences plus labeled documents
  - Label-Embedding Attentive Model [Wang et al. ACL 2018]: jointly embeds words and labels so that attention mechanisms can be employed to discover category distinctive words
- Learning embedding for lexical entailment: Help determine which terms belong to a category
  - Hyperbolic models (Poincaré, Lorentz, and hyperbolic cone)
    - Proven successful in graded lexical entailment detection
    - supervised and require hypernym-hyponym training pairs
- CatE jointly learns the word vector representation in the embedding space and its distributional specificity without requiring supervision, and simultaneously consider relatedness and specificity of words when retrieving category representative terms

Why New Approaches for Discriminative Topic Mining?

- Traditional topic models (e.g., LDA)
  - Use document-topic and topic-word distributions to model text generation process
    - The bag-of-words generation assumption: Each word is drawn independently from the topic-word distribution w.o. considering correlations between adjacent words
  - Make explicit probabilistic assumptions regarding the text generation mechanism, resulting in high model complexity and inflexibility
- Traditional word embedding approach (e.g., Word2Vec)
  - Effectively capture word semantic correlations by mapping words with similar local contexts closer in the embedding space
  - Not impose particular assumptions on the type of data distribution of the corpus and enjoy greater flexibility and higher efficiency
  - But word embeddings usually do not exploit document-level co-occurrences of words (i.e., global contexts) and also cannot naturally incorporate latent topics into the model without making topic-relevant generative assumptions

Modeling Text Generation: A Three-Step Process

- Step 1: Document d is generated conditioned on one of the n categories
  - Explicitly models the associations between each document and user-interested categories (i.e., topic assignment)
- Step 2: Each word w_i is generated conditioned on the semantics of the document d
  - Makes sure each word is generated in consistency with the semantics of its belonging document (i.e., global contexts
- Step 3: Surrounding words w_{i+j} in the local context window ($-h \le j \le h$) of w_i are generated conditioned on the semantics of the center word w_i
  - Models the correlations of adjacent words in the corpus (i.e., local contexts)
- Thus the likelihood of corpus generation conditioned on a specific set of user-interested categories C: (where c_d is the latent category of d)

$$P(\mathcal{D} \mid C) = \prod_{d \in \mathcal{D}} p(d \mid c_d) \prod_{w_i \in d} p(w_i \mid d) \prod_{\substack{w_{i+j} \in d \\ -h \le j \le h, j \neq 0}} p(w_{i+j} \mid w_i)$$

Derivation: Finding Words that Belong to Categories

$$P(\mathcal{D} \mid C) = \prod_{d \in \mathcal{D}} p(d \mid c_d) \prod_{w_i \in d} p(w_i \mid d) \prod_{\substack{w_{i+j} \in d \\ -h \le j \le h, j \neq 0}} p(w_{i+j} \mid w_i)$$

$$\mathcal{L} = -\sum_{d \in \mathcal{D}} \log p(d \mid c_d) \qquad (\mathcal{L}_{\text{topic}})$$

$$-\sum_{d \in \mathcal{D}} \sum_{w_i \in d} \log p(w_i \mid d) \qquad (\mathcal{L}_{\text{global}})$$

$$-\sum_{d \in \mathcal{D}} \sum_{w_i \in d} \sum_{\substack{w_{i+j} \in d \\ -h \le j \le h, j \neq 0}} \log p(w_{i+j} \mid w_i). \qquad (\mathcal{L}_{\text{local}}) \qquad (2)$$

- Take negative log-likelihood as our objective L:
- In Eq. (2), $p(w_i|d)$ and $p(w_{i+j}|w_i)$ are observable while $p(d \mid c_d)$ is latent
- To directly leverage the word level user supervisions (i.e., category names), a natural solution is to decompose $p(d|c_d)$ into word-topic distributions:

$$p(d \mid c_d) \propto p(c_d \mid d)p(d) \propto p(c_d \mid d) \propto \prod_{w \in d} p(c_d \mid w).$$

- Rewrite the first term in Eq. (2) by reorganizing the summation over categories instead of documents:

$$\mathcal{L}_{\text{topic}} = -\sum_{d \in \mathcal{D}} \log p(d \mid c_d) = -\sum_{c \in C} \sum_{w \in c} p(c \mid w) + \text{const.}$$

  - This becomes the category assignment of words, i.e., finding words that belong to the categories

Formulate It as an Embedding Learning Problem

- Formulate the optimization of the objective L as an embedding learning problem
  - Define the 3 probability expressions in Eq. (2) via log-linear models in the embedding space

$$p(c_i \mid w) = \frac{\exp(\boldsymbol{c}_i^\top \boldsymbol{u}_w)}{\sum_{c_j \in C} \exp(\boldsymbol{c}_j^\top \boldsymbol{u}_w)}, \quad (3)$$

$$p(w_i \mid d) = \frac{\exp(\boldsymbol{u}_{w_i}^\top \boldsymbol{d})}{\sum_{d' \in \mathcal{D}} \exp(\boldsymbol{u}_{w_i}^\top \boldsymbol{d}')}, \quad (4)$$

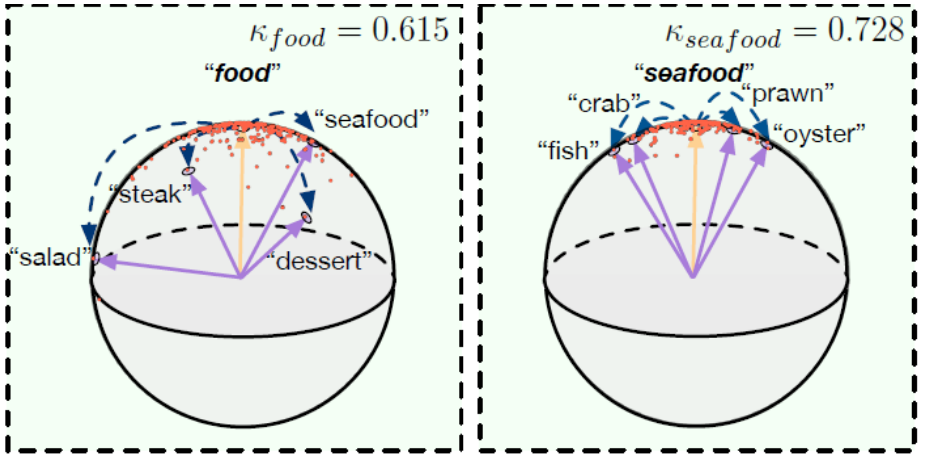
$$p(w_{i+j} \mid w_i) = \frac{\exp(\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w_{i+j}})}{\sum_{w' \in V} \exp(\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w'})}, \quad (5)$$

u_w: input vector representation of w (usually used as the word embedding)
v_w: output vector representation that serves as w's contextual representation
d: the document embedding
c_i: the category embedding

  - Eqs. (4) and (5) can be directly plugged into Eq. (2) to train word and document embeddings
  - Eq. (3) requires knowledge about the latent topic (i.e., the category that w belongs to) of a word w. Initially, we only know the user-provided category names, but with iterative topic mining, we will retrieve more terms under each category, gradually discovering the latent topic of more words

Retrieval of Category Representative Terms

- Retrieve category representative terms by jointly considering two separate aspects: Relatedness and specificity
- Constraint 1: w is semantically related to c
  - Simply requiring high cosine similarity between a candidate word embedding and the category embedding
- Constraint 2: w is semantically more specific than the category name of c
  - Improving the text embedding model by incorporating word specificity signals
- Word Distributional Specificity: Assume there is a scalar $\kappa_w \geq 0$ correlated with each word w indicating how specific the word meaning is
  - The bigger κ_w is, the more specific meaning w has, and the less varying contexts w appears in
  - Example: seafood vs. food



Jointly Learning Word Embedding and Distributional Specificity

- Modify Eqs. (4) and (5) to incorporate an additional learnable scalar κ_w for each word w, while constraining the embeddings to be on the unit hyper-sphere S^{p-1}

$$p(w_i \mid d) = \frac{\exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{d})}{\sum_{d' \in \mathcal{D}} \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{d'})}, \tag{7}$$

$$p(w_{i+j} \mid w_i) = \frac{\exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w_{i+j}})}{\sum_{w' \in V} \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w'})}, \tag{8}$$

$$s.t. \quad \forall w, d, c, \quad \|\boldsymbol{u}_w\| = \|\boldsymbol{v}_w\| = \|\boldsymbol{d}\| = \|\boldsymbol{c}\| = 1.$$

  - The unit norm constraints can be satisfied by simply normalizing the embedding vectors after each update. The κ_w learned is the distributional specificity of w.
- The contexts vectors are assumed to be generated from the vMF distribution with the center word vector u_{wi} as the mean direction and κ_{wi} as the concentration parameter
- Our model essentially learns both word embedding and word distributional specificity that maximize the probability of the context vectors getting generated by the center word's vMF distribution

Ranking Measure for Category Representative Words Retrieval

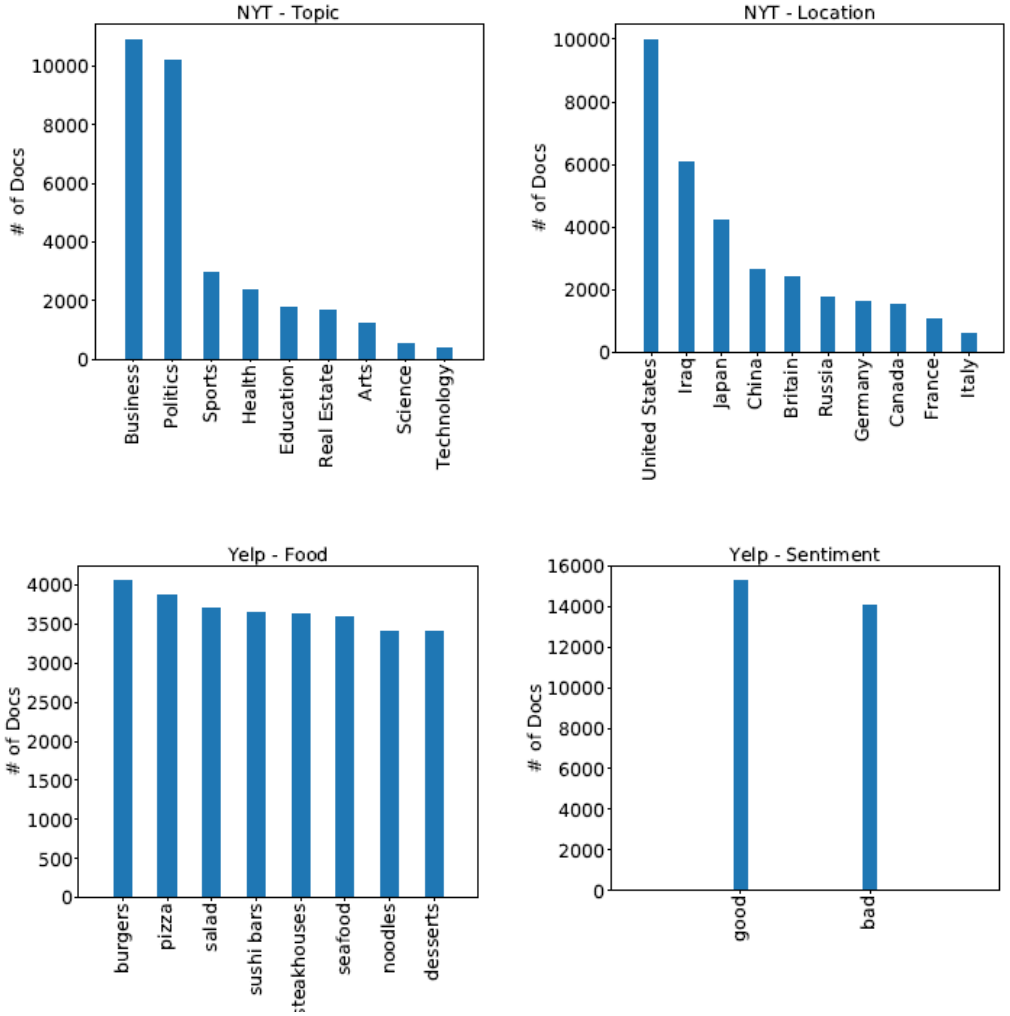
- A category representative word must have higher distributional specificity than the category name
- We also want to avoid selecting too specific terms as category representative words, which may appear fewer times in the corpus and suffer from lower embedding quality and higher variance due to insufficient training
- Among all the words that are more specific than the category name, we prefer words that (1) have high embedding cosine similarity with the category name, and (2) have low distributional specificity, which indicates wider semantic coverage

$$w = \arg\min_{w} \text{rank}_{sim}(w, c_i) \cdot \text{rank}_{spec}(w)$$
$$s.t. \quad w \notin \mathcal{S} \quad \text{and} \quad \kappa_w > \kappa_{c_i},$$

  - where $\text{rank}_{sim}(w, c_i)$ is the ranking of w by embedding cosine similarity with category c_i, i.e., $\cos(u_w, c_i)$, from high to low; $\text{rank}_{spec}(w)$ is the ranking of w by distributional specificity, i.e., κ_w, from low to high

Experiment Setting: Datasets and Preprocessing

- Two datasets:
  - New York Times annotated corpus (NYT)
    - Two categories: topic and location
  - Recently released Yelp Dataset Challenge (Yelp)
    - Two categories: food type and sentiment
- Preprocessing:
  - Use AutoPhrase to extract quality phrases, treated as single words during embedding training
  - Hyperparameter setting:
    - word embedding dimension p = 100
    - local context window size h = 5
    - number of negative samples k = 5
    - training iterations on the corpus max_iter = 10



Dataset stat: # of docs by category name

Compared Baselines and Evaluation Metrics

- Baselines: including traditional topic modeling, seed-guided topic modeling and embedding-based topic modeling
- Input: # of topics N. We vary N in [n, 2n, . . . , 10n] where n is the actual number of categories and report the best performance of the method.
- Compared methods: LDA [NIPS'08], Seeded LDA [EACL'12], TWE [AAAI'15], Anchored CorEx [TACL'17], Labeled ETM [Dieng, Ruiz, Blei'19]
- Evaluation Metrics
  - Topic coherence (TC): Term coherence inside each topic (avg. normalized PMI of two words randomly drawn from the same document)
  - Mean accuracy (MACC): The proportion of retrieved top words that actually belong to the category defined by user-provided category names

$$\mathrm{TC} = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{45}\sum_{i=1}^{10}\sum_{j=i+1}^{10}-\frac{\log\frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{\log P(w_i,w_j)}$$

$P(w_i, w_j)$: probability of w_i and w_j co-occurring in a document

$$\mathrm{MACC} = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{10}\sum_{i=1}^{10}\mathbb{1}(w_i \in c_k)$$

$1(w_i \in c_k)$: indicator function of whether w_i belongs to category c_k

Performance Study on Discriminative Topic Mining

- Quantitative comparison

Methods	NYT-Location		NYT-Topic		Yelp-Food		Yelp-Sentiment	
	TC	MACC	TC	MACC	TC	MACC	TC	MACC
LDA	0.007	0.489	0.027	0.744	-0.033	0.213	-0.197	0.350
Seeded LDA	0.024	0.168	0.031	0.456	0.016	0.188	0.049	0.223
TWE	0.002	0.171	-0.011	0.289	0.004	0.688	-0.077	0.748
Anchored CorEx	0.029	0.190	0.035	0.533	0.025	0.313	0.067	0.250
Labeled ETM	0.032	0.493	0.025	0.889	0.012	0.775	0.026	0.852
CatE	**0.049**	**0.972**	**0.048**	**0.967**	**0.034**	**0.913**	**0.086**	**1.000**

- Qualitative comparison
  - @the next page
- Discussion
  - LDA retrieves reasonably good topics relevant to category names but need careful manual selection of topics
  - Four guided topic modeling baselines alleviate the burden of manual selection
  - Seeded LDA and Anchored CorEx may get semantically irrelevant results
  - TWE and Labeled ETM employ distributed word representations, obtaining semantically relevant terms but may not actually belong to the corresponding category (e.g., “Germany” and “Europe” under “Britain”)

Comparative Evaluation of Discriminative Topic Mining

Methods	NYT-Location		NYT-Topic		Yelp-Food		Yelp-Sentiment	
	britain	canada	education	politics	burger	desserts	good	bad
LDA	company (×)	percent (×)	school	campaign	fatburger	ice cream	great	valet (×)
	companies (×)	economy (×)	students	clinton	dos (×)	chocolate	place (×)	peter (×)
	british	canadian	city (×)	mayor	liar (×)	gelato	love	aid (×)
	shares (×)	united states (×)	state (×)	election	cheeseburgers	tea (×)	friendly	relief (×)
	britain	trade (×)	schools	political	bearing (×)	sweet	breakfast	rowdy
Seeded LDA	british	city (×)	state (×)	republican	like (×)	great (×)	place (×)	service (×)
	industry (×)	building (×)	school	political	fries	like (×)	great	did (×)
	deal (×)	street (×)	students	senator	just (×)	ice cream	service (×)	order (×)
	billion (×)	buildings (×)	city (×)	president	great (×)	delicious (×)	just (×)	time (×)
	business (×)	york (×)	board (×)	democrats	time (×)	just (×)	ordered (×)	ordered (×)
TWE	germany (×)	toronto	arts (×)	religion	burgers	chocolate	tasty	subpar
	spain (×)	osaka (×)	fourth graders	race	fries	complimentary (×)	decent	positive (×)
	manufacturing (×)	booming (×)	musicians (×)	attraction (×)	hamburger	green tea (×)	darned (×)	awful
	south korea (×)	asia (×)	advisors	era (×)	cheeseburger	sundae	great	crappy
	markets (×)	alberta	regents	tale (×)	patty	whipped cream	suffered (×)	honest (×)
Anchored CorEx	moscow (×)	sports (×)	republican (×)	military (×)	order (×)	make (×)	selection (×)	did (×)
	british	games (×)	senator (×)	war (×)	know (×)	chocolate	prices (×)	just (×)
	london	players (×)	democratic (×)	troops (×)	called (×)	people (×)	great	came (×)
	german (×)	canadian	school	baghdad (×)	fries	right (×)	reasonable	asked (×)
	russian (×)	coach	schools	iraq (×)	going (×)	want (×)	mac (×)	table (×)
Labeled ETM	france (×)	canadian	higher education	political	hamburger	pana	decent	horrible
	germany (×)	british columbia	educational	expediency (×)	cheeseburger	gelato	great	terrible
	canada (×)	britain (×)	school	perceptions (×)	burgers	tiramisu	tasty	good (×)
	british	quebec	schools	foreign affairs	patty	cheesecake	bad (×)	awful
	europe (×)	north america (×)	regents	ideology	steak (×)	ice cream	delicious	appallingly
CatE	england	ontario	educational	political	burgers	dessert	delicious	sickening
	london	toronto	schools	international politics	cheeseburger	pastries	mindful	nasty
	britons	quebec	higher education	liberalism	hamburger	cheesecakes	excellent	dreadful
	scottish	montreal	secondary education	political philosophy	burger king	scones	wonderful	freaks
	great britain	ottawa	teachers	geopolitics	smash burger	ice cream	faithful	cheapskates

Experiments on Weakly-Supervised Document Classification

- Weakly-supervised document classification: uses category names or a set of keywords from each category instead of human annotated documents to train a classifier
- We use WeSTClass [CIKM'18] as the weakly-supervised classification model, but replace Word2Vec with different embedding models as input features
- Compared Methods
  - Word2Vec: a predictive word embedding model that learns distributed representations by maximizing the probability of using the center word to predict its local context words or in the opposite way.
  - GloVe: learns word embedding by factorizing a global word-word co-occurrence matrix where the co-occurrence is defined upon a fix-sized context window
  - fastText [TACL'17]: extension of by incorporating subword information
  - BERT [NAACL'19]: A state-of-the-art pretrained language model (using bidirectional Transformers) that provides contextualized word representations

Results on Weakly-Supervised Document Classification

Embedding	**NYT**-Location		**NYT**-Topic		**Yelp**-Food		**Yelp**-Sentiment	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word2Vec	0.533	0.467	0.588	0.695	0.540	0.528	0.723	0.715
GloVe	0.521	0.455	0.563	0.688	0.515	0.503	0.720	0.711
fastText	0.543	0.485	0.575	0.693	0.544	0.529	0.738	0.743
BERT	0.301	0.288	0.328	0.451	0.330	0.404	0.695	0.674
CatE	**0.655**	**0.613**	**0.611**	**0.739**	**0.656**	**0.648**	**0.838**	**0.836**

Weakly-supervised document classification evaluation based on WeSTClass model

- Unsupervised embeddings (Word2Vec, GloVe and fastText) do not really have notable differences as word representations to WeSTClass
- BERT is not suitable for classification without sufficient training data, probably because BERT embedding has higher dimensionality, which might require stronger supervision signals to tune
- CatE outperforms all unsupervised embeddings but only marginally for NYT-topic
- Weakly-supervised classification: Under label scarcity scenarios, using word-level supervision only can bring significant improvements to weakly-supervised models

Unsupervised Lexical Entailment Direction Identification

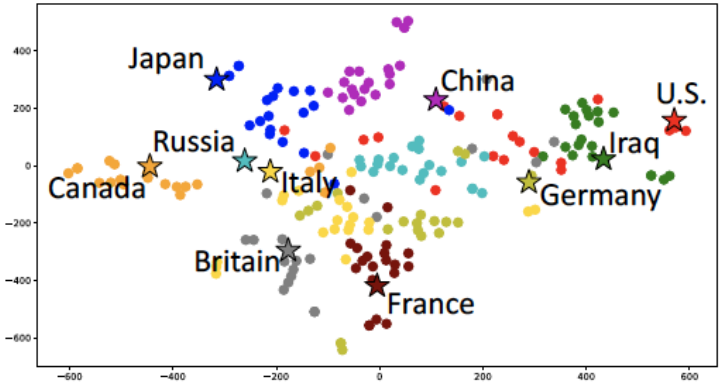
- Lexical entailment (LE) refers to the "type-of" (aka. hyponymy-hypernymy) relation
  - Discriminate hypernymy from other relations (detection)
  - Identify which one is hyponymy (direction identification)
- Compared Methods: focus on unsupervised methods for LE direction identification
  - Frequency: "hypernyms are more frequent than hyponyms in the corpus"
  - SLQS [EACL'14]: measures the generality of a term via the entropy of its statistically most prominent context
  - Vec-Norm: A general term tends to have a lower embedding norm (it co-occurs with many different terms and its vector is dragged from different directions)
  - HyperVec [EMNLP'17]: arranging the word embedding training order corresponding to the hypernym–hyponym distributional hierarchy

Methods	Frequency	SLQS	Vec-Norm	HyperVec	CatE
Accuracy	0.659	0.871	0.562	0.918	**0.923**

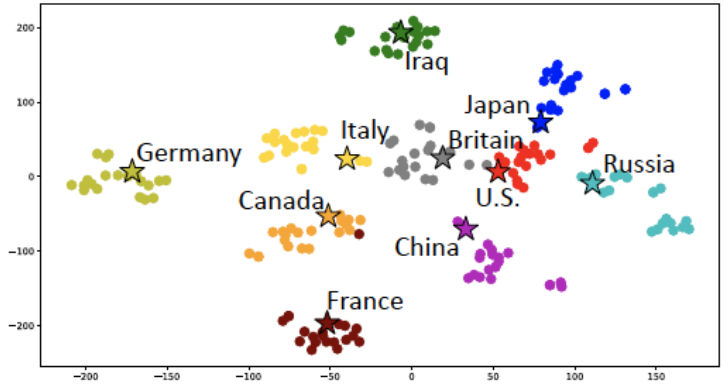
Accuracy for hypernymy direction identification: "training" on the latest Wikipedia dump (w. 2.4 billion tokens)

Case Study I: Visualize the Trained Embeddings

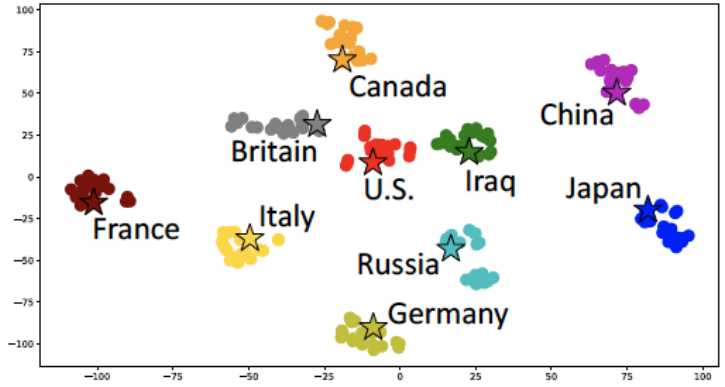
- Visualize the embeddings trained on NYT-Location
  - star (category embeddings), points (phrases)



(a) Epoch 1



(b) Epoch 3



(c) Epoch 5

- The categories become well-separated in the embedding space during training
  - Category representative words gather around their corresponding category in the embedding space
  - Other semantically similar words move towards their belonging categories as well

Case Study 2: Effect of Distributional Specificity

- Coarse-to-fine topic presentation on NYT-Topic

Range of κ	Science ($\kappa_c = 0.539$)	Technology ($\kappa_c = 0.566$)	Health ($\kappa_c = 0.527$)
$\kappa_c < \kappa < 1.25\kappa_c$	scientist, academic, research, laboratory	machine, equipment, devices, engineering	medical, hospitals, patients, treatment
$1.25\kappa_c < \kappa < 1.5\kappa_c$	physics, sociology, biology, astronomy	information technology, computing, telecommunication, biotechnology	mental hygiene, infectious diseases, hospitalizations, immunizations
$1.5\kappa_c < \kappa < 1.75\kappa_c$	microbiology, anthropology, physiology, cosmology	wireless technology, nanotechnology, semiconductor industry, microelectronics	dental care, chronic illnesses, cardiovascular disease, diabetes
$\kappa > 1.75\kappa_c$	national science foundation, george washington university, hong kong university, american academy	integrated circuits, assemblers, circuit board, advanced micro devices	juvenile diabetes, high blood pressure, family violence, kidney failure

- The table below lists the most similar words/phrases with each category (measured by embedding cosine similarity) from different ranges of distributional specificity
- When κ is smaller, the retrieved words have wider semantic coverage
- In our model design, if not imposing constraints on the distributional, the retrieved words might be too general and do not belong to the category

Application: Support Multi-Dimensional Text Analysis

Cube Demo | Time: 2014-07 | Category: infrastructure | PROVINCE NAME | UPDATE | CURRENT: CHERKASY

IMAGE & TOP-K KEYWORDS & SUMMARY

ELATED IMAGE AND KEYWORDS.

Ukraine-Russia Conflicts: MH17 Shot-Down

‹ PREV | NEXT ›

SHOT DOWN | PASSENGER JET

PLANE CRASH | MISSILE FIRED

BLACK BOX | CIVIL AVIATION

TOP PRIORITY | AIR TRAFFIC CONTROL

AIR TRAFFIC | REBEL LEADER

Malaysia Airlines flight MH17 crash: 'Nine Britons, 23 Americans and 80 children' feared dead after Boeing passenger jet is 'shot down' near Ukraine-Russia border Rescuers stand on the site of the crash of a Malaysian airliner near the town of Shaktarsk, in rebel-held east Ukraine Nine Britons, 23 US citizens and 80 children are reported to be among the 298 people killed when a Malaysia Airlines jet crashed near the eastern Ukraine border on Thursday.

Analysis of Russia-Ukraine Conflicts

Category representative phrases generated automatically

category names and three examples from the experts

POLITICAL	MILITARY	ECONOMIC	SOCIAL	INFORMATION	CIVILIAN
Political power	Military forces	Employment	Demographic	Infowars	Urban areas
Dictator	Infantry	Economic activity	Ethnic	Information warfare	Residential area
Anarchy	Insurgents	Market	Population	Radio	Utilities
Pro government	Combatants	Finance	Language	Information security	Transportation
Neo nazi	National guard	European union	Ethnic russians	Ekho moskvy	Nuclear power plants
Viktor yanukovych	Armored vehicles	Foreign policy	Soviet union	Ukraine http empr	Power plants
Right sector	Special forces	Sergei ivanov	Western ukraine	Social media	Nuclear fuel
Pro russian	Self defense	Interior ministry	Russian language	News media	Crash site
Opposition politicians	Armored personnel	Economic sanctions	Police state	Novaya gazeta	Civil aviation
Maidan movement	Pro russian separatists	Rinat akhmetov	Anglo zionist empire	Ria novosti	Surface to air missile
Pro western	Donetsk oblast	Billion dollars	Maidan supporters	Rfe rl	Contaminated water
Kulikovo pole	Heavy fighting	Right sector	The vast majority	Mainstream media	Main entrance
Communist party	Peoples militia	Closer ties	Social media	Main stream	Emergency services
Civil war	Automatic rifles	Magnitsky act	Martial law	Intelligence community	Drinking water

Cube Demo | Dataset: Hong Kong Protest | Time: 2019.08.12-08.18 | Category: transportation | City: Islands

IMAGE & TOP-K KEYWORDS & SUMMARY

IT SHOWS THE RELATED IMAGE AND KEYWORDS.

ALLEGEDLY SHOT

EYE PATCHES

TEAR GAS INSIDE

PATCHES

AIRPORTS

AIRPORT SECURITY

CHASING PROTESTERS

CHARGED PROTESTERS

BEANBAG ROUND

NEWS FOOTAGE

Demonstrators don eye patches at Lantau Island hub, one of the world's busiest international airports, in anger that a girl allegedly shot with a police beanbag round could lose an eye \n Sit-in comes after night of escalated violence inside subway stations \n Demonstrators don eye patches at Lantau Island hub, one of the world's busiest international airports, in anger that a girl allegedly shot with a police beanbag round could lose an eye.

Multidimensional Analysis of News Data : HK Protests

Top-k phrases generated by CatE

Summary: extracted w. top-k phrases

Analysis of Hong Kong Protests

Category representative phrases generated automatically

IT SHOWS RELEVANT WORDS OF DIFFERENT CATEGORIES;

category names and three examples from the experts

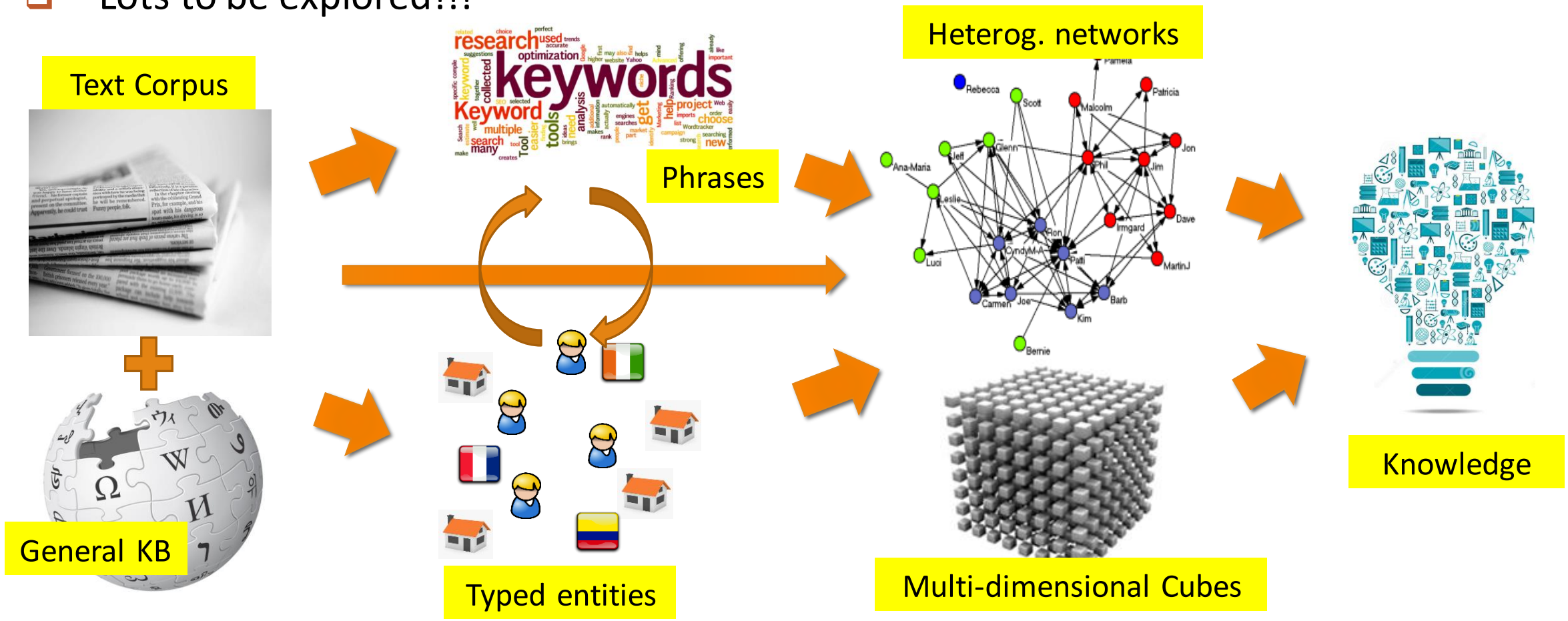
POLITICAL	POLICE	ECONOMIC	INFORMATION	INFRASTRUCTURE
pro democracy	tear gas	financial crisis	cbc news	hong kong university
pro beijing	hong kong police	economic downturn	cbs news	transportation
hong kong government	riot police	economic growth	fox news	international aiport
Chief executive	Water cannon	Infrastructure	Chinese state media	Mass transit railway
Mainland china	Pepper spray	Real estate	Bbc news	Lantau link
Pro establishment	Petrol bombs	Affordable housing	Global times	Flight cancellations
Mainland chinese	Hong kong government	Trade war	News media	Victoria harbour
Chief executive carrie lam	Beanbag rounds	The united states	Sina weibo	Rail operator
Carrie lam	Firing tear gas	Financial secretary	Internet censorship	Busiest airports
The chinese government	Tsuen wan	Global financial	Local media	Public transport

Outline

- Text Mining, Text Similarity and Text Embedding
- Unsupervised word embedding
  - Context-free representation:
    - Euclidean Embedding: Word2Vec, GloVe, fastText
      - Local-Corpus Based Embedding
    - Hyperbolic Embedding: Poincaré and Lorentz Embedding
    - Spherical Text Embedding: JoSE (Joint Spherical Text Embedding)
  - Contextualized representation: ELMo, BERT, XLNet
- CatE: Category-Name Guided Text Embedding for Topic Mining
- Looking Forward

Looking Forward: Structural Mining of Massive Text Data

- From big data to big knowledge
  - A key problem: **Structural mining of massive text data**
  - Lots to be explored!!!



Summary

- Big data and big network challenge: The curse of high dimensionality
- Embedding: A powerful new approach
  - Mapping high dimensional data into low-dimensional vector space
- Tons of research work on embedding (we focus on text analysis)
  - Word embedding from massive text data: Word2Vec (Mikolov et al. NIPS'2013)
  - Spherical text embedding (Meng et al. NeurIPS'19)
- Broad applications
  - Word/phrase similarity analysis, document clustering, classification, …
  - Discriminative topic mining
- Future work
  - A lot of exciting work to come

References (I)

- A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von mises-fisher distributions. Journal of Machine Learning Research, 6:1345–1382, 2005
- K. Batmanghelich, A. Saeedi, K. Narasimhan, and S. J. Gershman. Nonparametric spherical topic modeling with word embeddings. In ACL, 2016
- P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. TACL, 2017
- S. Bonnabel. Stochastic gradient descent on Riemannian manifolds. IEEE Transactions on Automatic Control, 58:2217–2229, 2013
- K. Cho, B. van Merrienboer, Çaglar Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In EMNLP, 2014
- J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. ***NAACL* (2019)**
- B. Dhingra, C. J. Shallue, M. Norouzi, A. M. Dai, and G. E. Dahl. Embedding text in hyperbolic spaces. In TextGraphs@NAACL-HLT, 2018
- Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2019. Topic Modeling in Embedding Spaces. ArXiv abs/1907.04907 (2019)
- O.-E. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In ICML, 2018

References (II)

- H. Gui, Q. Zhu, L. Liu, A. Zhang, J. Han. Expert Finding in Heterogeneous Bibliographic Networks with Locally-trained Embeddings. ArXiv: 1803.03370 (2018)
- Y. Kim. Convolutional neural networks for sentence classification. In EMNLP, 2014
- S. Kumar and Y. Tsvetkov. Von mises-fisher loss for training sequence to sequence models with continuous outputs. In ICLR, 2019
- Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In ICML, 2014
- Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical Word Embeddings. In AAAI.
- Y. Meng, J. Huang, G. Wang, C. Zhang, H. Zhuang, L. Kaplan and J. Han, "Spherical Text Embedding", **NeurIPS'19**
- Y. Meng, J. Shen, C. Zhang, and J. Han. Weakly-supervised neural text classification. In CIKM, 2018
- Y. Meng, J. Shen, C. Zhang, and J. Han. Weakly-supervised hierarchical text classification. In AAAI, 2019
- Y Meng, J. Huang, G. Wang, Z. Wang, C. Zhang, Y. Zhang and J. Han, "Discriminative Topic Mining via Category-Name Guided Text Embedding", **WWW'20**
- T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013

References (III)

- T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality, **NIPS'13**
- M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In NIPS, 2017
- M. Nickel and D. Kiela. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In ICML, 2018
- J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In EMNLP, 2014
- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M.P., Clark, C., Lee, K., & Zettlemoyer, L.S. (2018). Deep contextualized word representations. NAACL.
- J. Tang, M. Qu, and Q. Mei. PTE: Predictive text embedding through large-scale heterogeneous text networks. In KDD, 2015
- J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding", WWW'15
- F. Tao, C. Zhang, X. Chen, M. Jiang, T. Hanratty, L. M. Kaplan, and J. Han. Doc2cube: Allocating documents to text cube without labeled data. In ICDM, 2018
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. NeurIPS'19