Part I: Overview of Text Embedding Methods

KDD 2020 Tutorial

Embedding-Driven Multi-Dimensional Topic Mining and Text Analysis

Yu Meng, Jiaxin Huang, Jiawei Han

Computer Science, University of Illinois at Urbana-Champaign

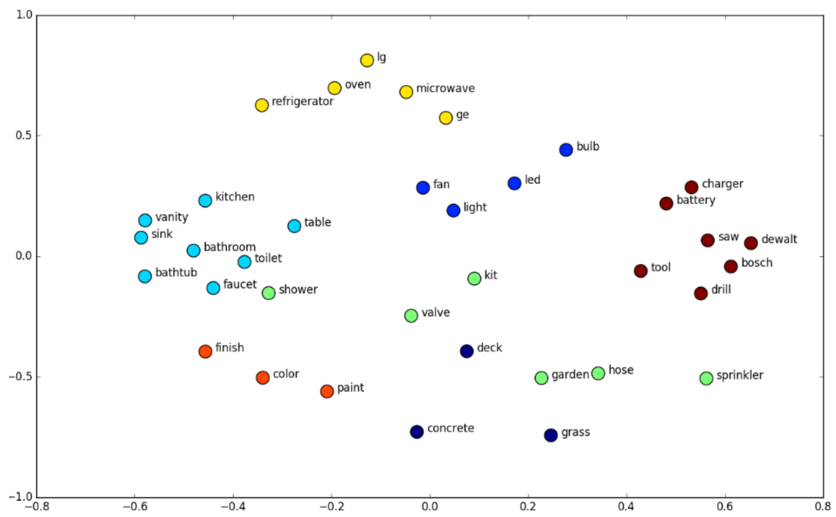
August 23, 2020

Outline

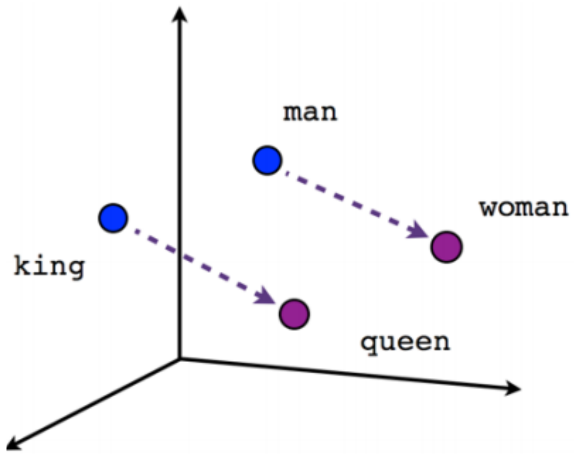
- ❑ Introduction to text embeddings
- ❑ Local context-based word embeddings
- ❑ Joint local and global context-based text embeddings
- ❑ Deep contextualized embeddings via neural language models
- ❑ Extend unsupervised embeddings to incorporate weak supervision

Introduction to Text Embeddings

- A milestone in NLP and ML:
  - Unsupervised learning of text representations—No supervision needed
  - Embed one-hot vectors into lower-dimensional 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)



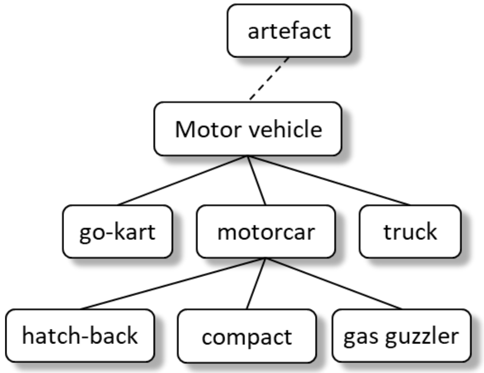
Word Similarity



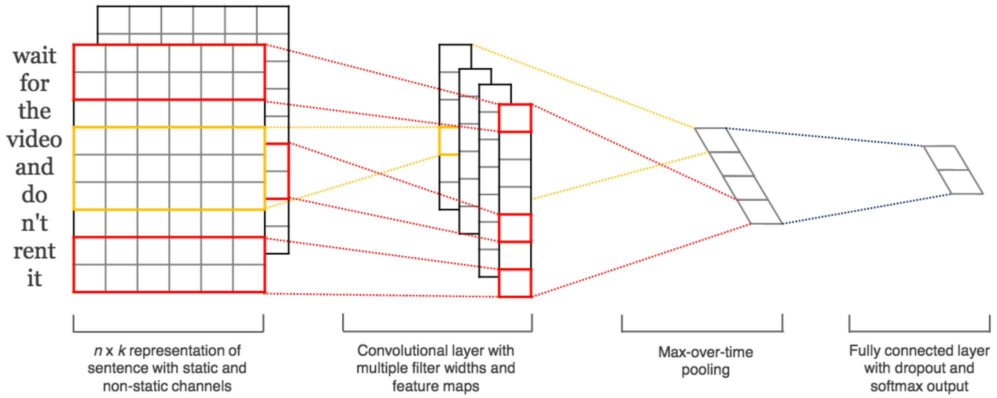
Word Analogy

Introduction to Text Embeddings

- Text embeddings can be used in a lot of downstream applications
  - Word/token/entity-level tasks
    - Keyword extraction/clustering
    - Taxonomy construction
  - Document/paragraph-level tasks
    - Document classification/clustering/retrieval
    - Question answering/text summarization



Taxonomy Construction



Document Classification

Outline

- Introduction to text embeddings
- Local context-based word embeddings
    - Euclidean space: Word2Vec, GloVe, fastText
    - Hyperbolic space: Poincaré embeddings
- Joint local and global context-based text embeddings
- Deep contextualized embeddings via neural language models
- Extend unsupervised embeddings to incorporate weak supervision

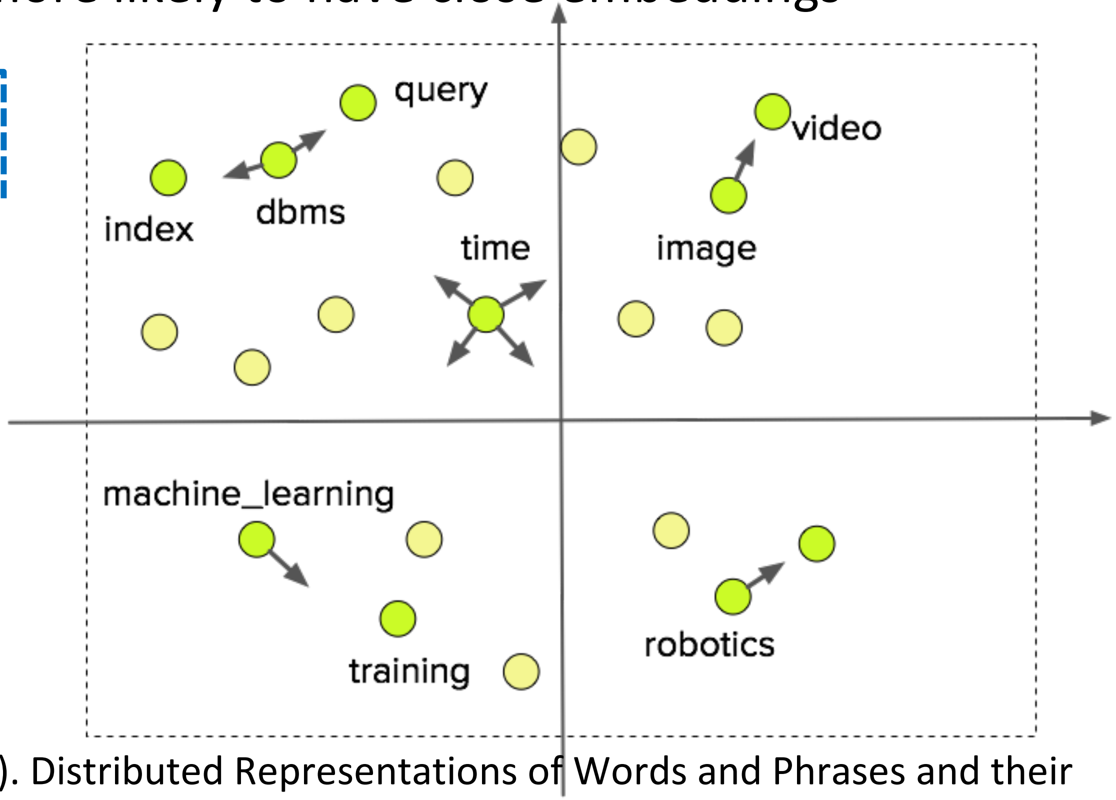
Word2Vec

- Local context-based word embedding learning pushes together terms that share same or similar **local contexts**
- For example, 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**

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

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

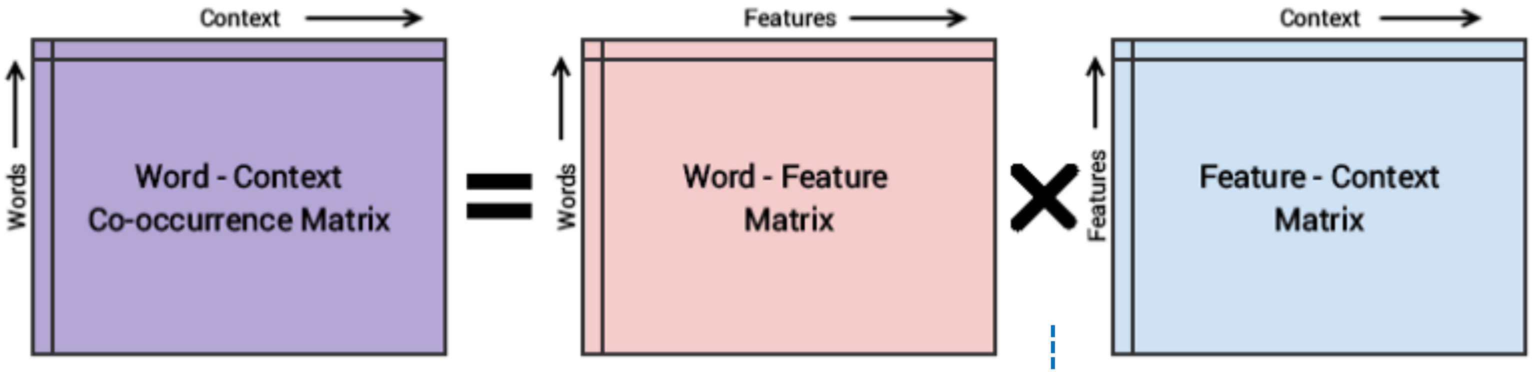


Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. NIPS.

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

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



Sparse, high dimensional

Low dimensional representation

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

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

$$p(w_O|w_I) = \frac{\exp\left(\boxed{{v'_{w_O}}^{\top} v_{w_I}}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^{\top} v_{w_I}\right)} \dashrightarrow \sum_{g \in \mathcal{G}_w} \boxed{\mathbf{z}_g^{\top}} \mathbf{v}_c.$$

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.

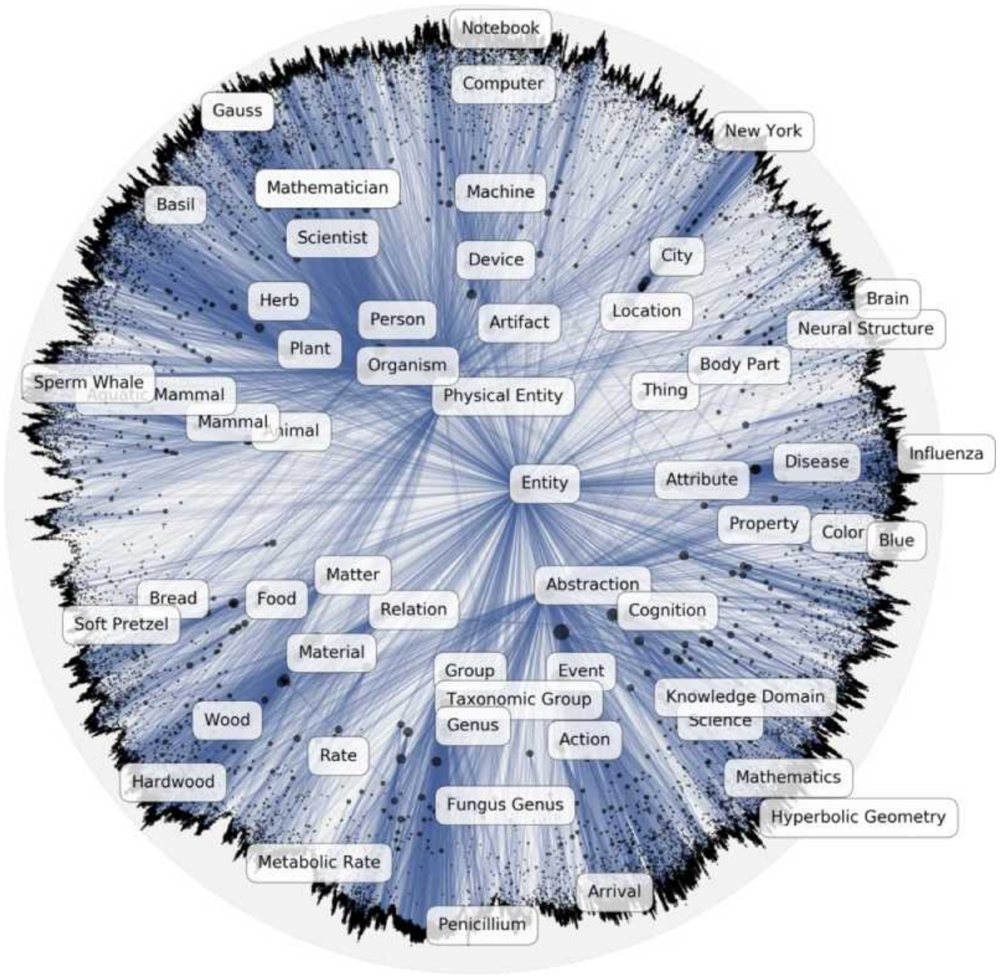
Outline

- Introduction to text embeddings
- Local context-based word embeddings
  - Euclidean space: Word2Vec, GloVe, fastText
  - Hyperbolic space: Poincaré embeddings
- Joint local and global context-based text embeddings
- Deep contextualized embeddings via neural language models
- Extend unsupervised embeddings to incorporate weak supervision

Hyperbolic Embedding: Poincaré embedding

- Why non-Euclidean embedding space?
  - Data can have specific structures that Euclidean-space models struggle to capture
- The hyperbolic space
  - Continuous version of trees
  - Naturally equipped to model hierarchical structures
- Poincaré embedding
  - Learn 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.

Texts in Hyperbolic Space: Poincaré GloVe

- GloVe in hyperbolic space
- Motivation: latent hierarchical structure of words exists among text
  - Hypernym-hyponym
  - Textual entailment
- Approach: use hyperbolic kernels!
- Effectively model generality/specificity

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

Hyperbolic metric

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

General

Specific

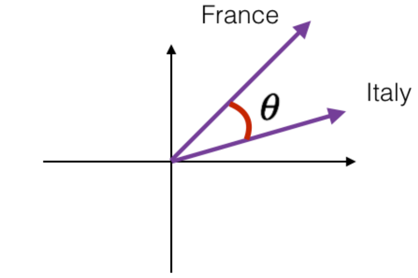
Tifrea, A., Bécigneul, G., & Ganea, O. (2019). Poincaré GloVe: Hyperbolic Word Embeddings. ICLR.

Outline

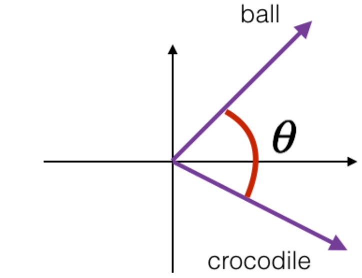
- Introduction to text embeddings
- Local context-based word embeddings
- Joint local and global context-based text embeddings
  - Spherical space: JoSE
- Deep contextualized embeddings via neural language models
- Extend unsupervised embeddings to incorporate weak supervision

Directional Analysis for Text Embeddings

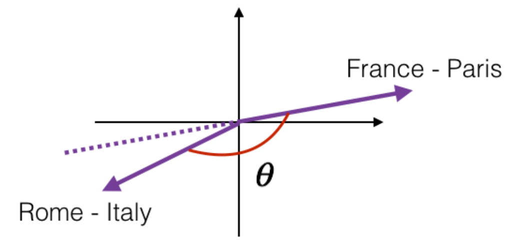
- How to use text embeddings? Mostly directional similarity (i.e., cosine similarity)
  - Word similarity is derived using cosine similarity



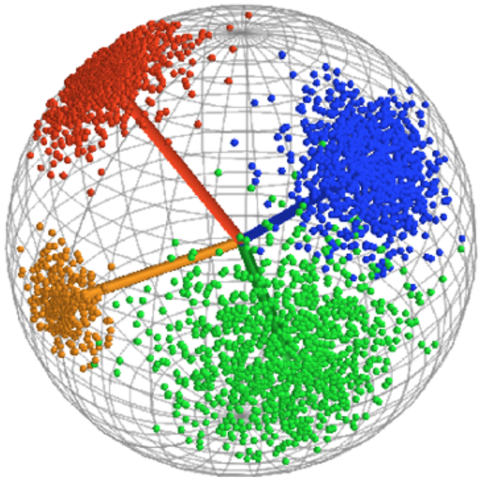
France and Italy are quite similar
θ is close to 0°
$\cos(\theta) \approx 1$



ball and crocodile are not similar
θ is close to 90°
$\cos(\theta) \approx 0$



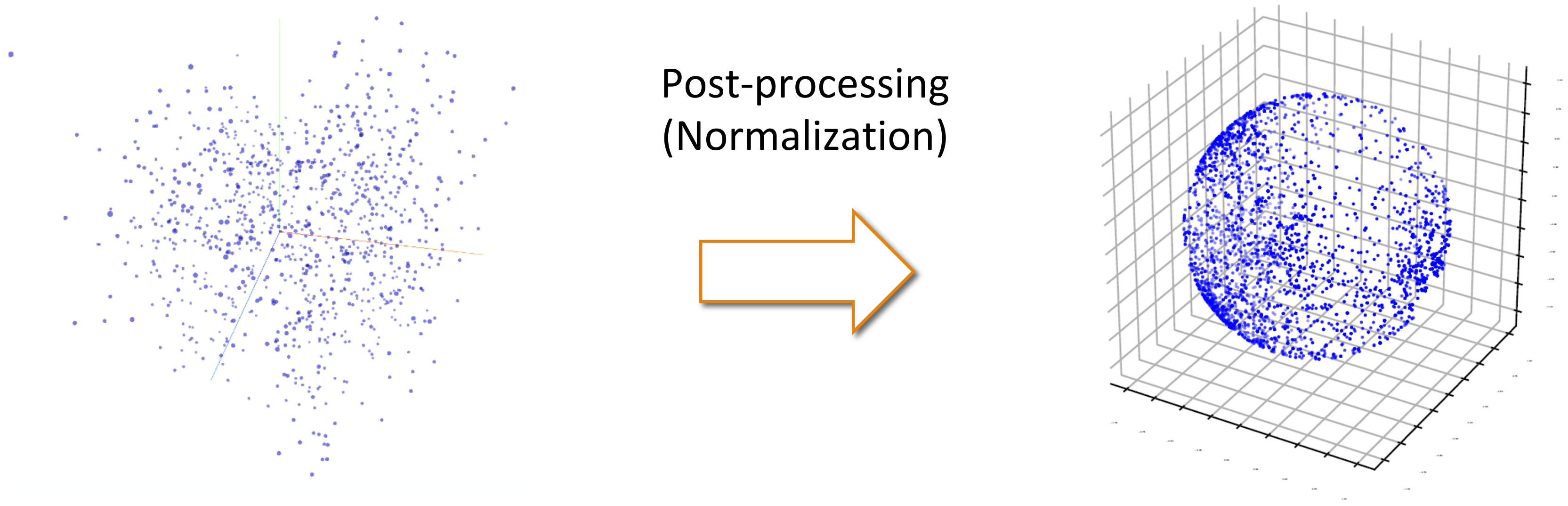
the two vectors are similar but opposite
the first one encodes (city - country)
while the second one encodes (country - city)
θ is close to 180°
$\cos(\theta) \approx -1$



  - Better clustering performances when embeddings are normalized and spherical clustering algorithms are used (Spherical K-means)
  - Vector direction is what actually matters!

Motivation

- Issues with previous word embedding frameworks:
  - Although directional similarity has shown effective for various applications, previous embeddings (e.g. Word2Vec, GloVe, fastText) are trained in the Euclidean space
  - A gap between training space and usage space: Trained in Euclidean space but used on sphere



Embedding Training in Euclidean Space

Embedding Usage on the Sphere
(Similarity, Clustering, etc.)

Motivation

- What is the consequence of 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

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

Word2Vec

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

GloVe

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

fastText

Motivation

- What is the consequence of the inconsistency between word embedding training and usage space?
  - The objective we optimize during training is not really the one we use
  - E.g. 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 testset)
  - 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*
Training	Dot Product	5.284	$<$	6.287
Usage	Cosine Similarity	0.637	$>$	0.628

Inconsistency

Motivation

- Apart from the training/usage space inconsistency issue, previous embedding frameworks only leverage **local contexts** to learn word representations
  - Local contexts can only partly define word semantics in unsupervised word embedding learning

> 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 …

Local contexts of "harmful"

Spherical Text Embedding

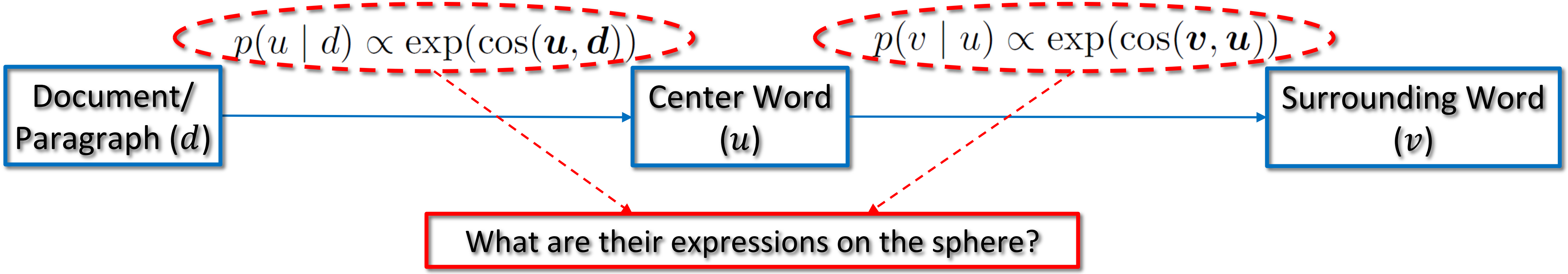
- We design a generative model on the sphere that follows how humans write articles:
  - We first have a general idea of the paragraph/document, and then start to write down each word in consistent with not only the paragraph/document, but also the surrounding words
  - Assume a two-step generation process:

$$\text{Document/ Paragraph } (d) \xrightarrow{p(u \mid d) \propto \exp(\cos(\boldsymbol{u}, \boldsymbol{d}))} \text{Center Word } (u) \xrightarrow{p(v \mid u) \propto \exp(\cos(\boldsymbol{v}, \boldsymbol{u}))} \text{Surrounding Word } (v)$$

Meng, Y., Huang, J., Wang, G., Zhang, C., Zhuang, H., Kaplan, L.M., & Han, J. (2019). Spherical Text Embedding. NeurIPS.

Spherical Text Embedding

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



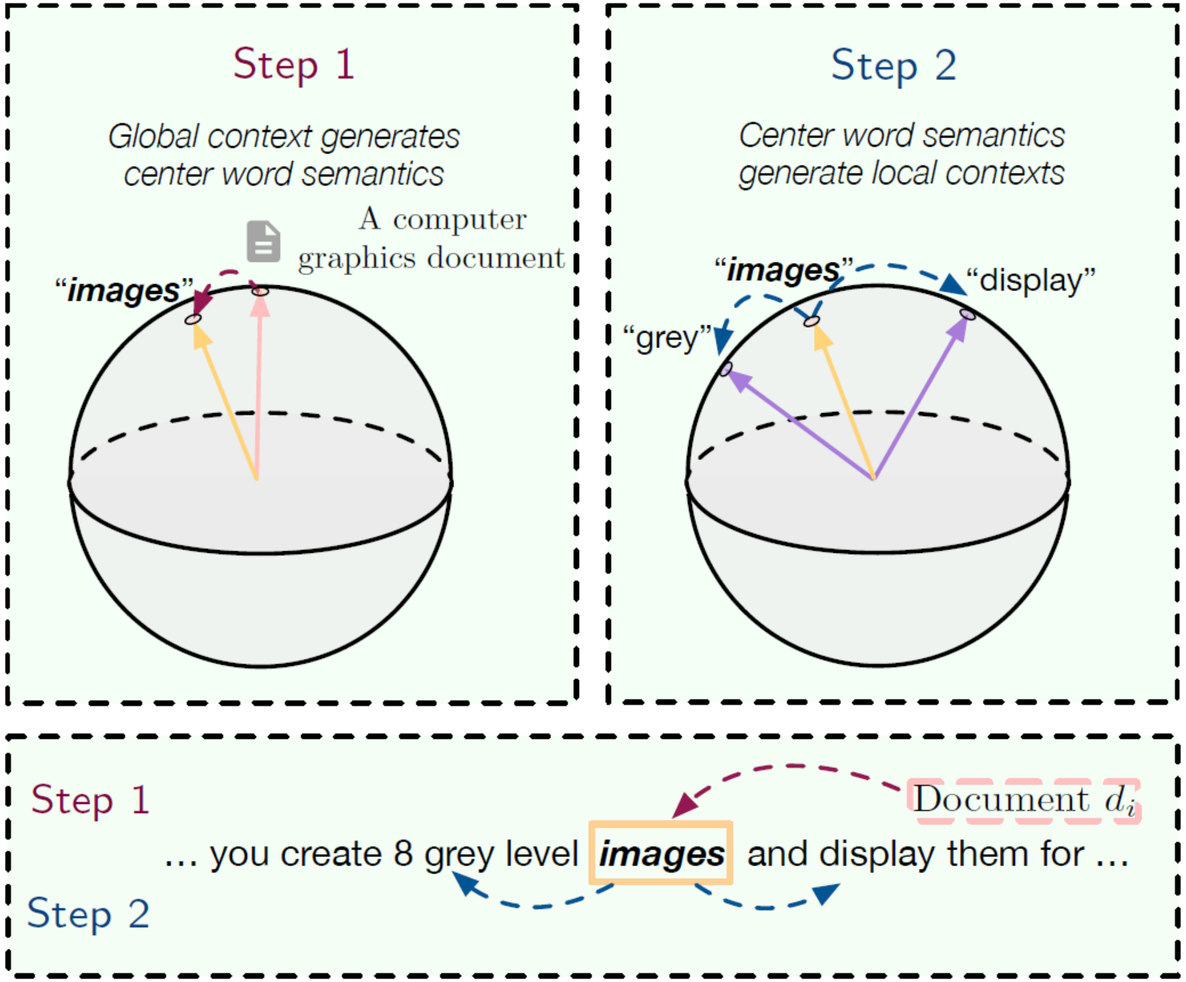
- We prove a theorem connecting the above generative model with a spherical probability distribution:

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.*,

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

Spherical Text Embedding

- Understanding the spherical generative model



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:

$$\begin{aligned}\mathcal{L}_{\text{joint}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{d}) = \max \Big(0, m - &\boxed{\log \left(c_p(1) \exp(\cos(\boldsymbol{v}, \boldsymbol{u})) \cdot c_p(1) \exp(\cos(\boldsymbol{u}, \boldsymbol{d})) \right)} \quad \text{Positive Sample} \\ &+ \boxed{\log \left(c_p(1) \exp(\cos(\boldsymbol{v}, \boldsymbol{u}')) \cdot c_p(1) \exp(\cos(\boldsymbol{u}', \boldsymbol{d})) \right)} \Big) \quad \text{Negative Sample} \\ = \max \left(0, m - \cos(\boldsymbol{v}, \boldsymbol{u}) - \cos(\boldsymbol{u}, \boldsymbol{d}) + \cos(\boldsymbol{v}, \boldsymbol{u}') + \cos(\boldsymbol{u}', \boldsymbol{d}) \right),&\end{aligned}$$

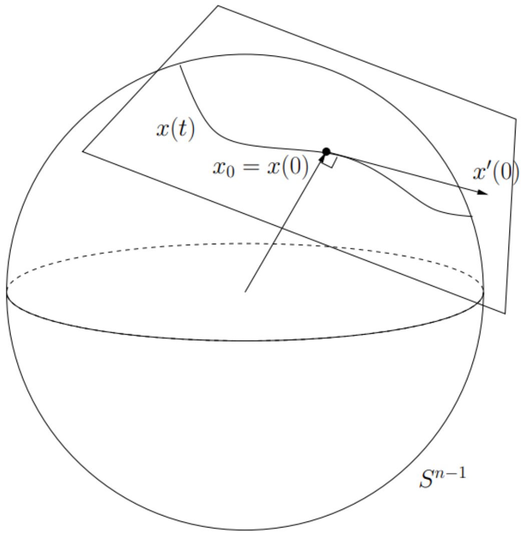
Optimization on the Sphere

- 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:

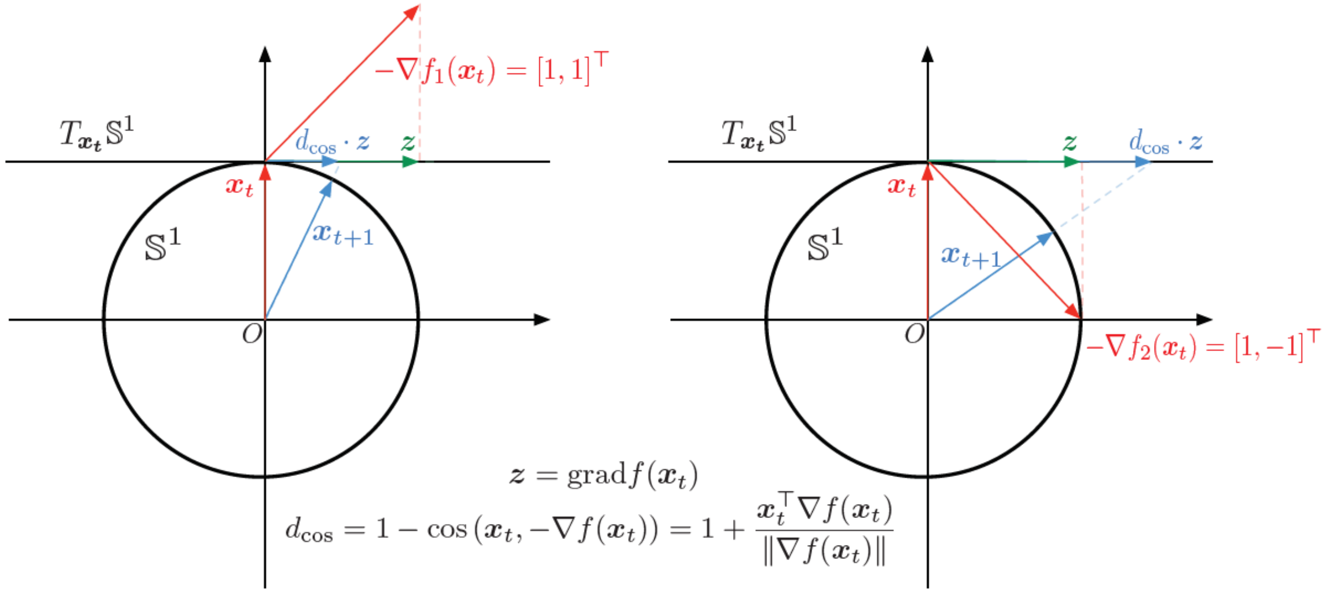
$$\boldsymbol{x}_{t+1} = \exp_{\boldsymbol{x}_t}\left(-\eta_t \operatorname{grad} f(\boldsymbol{x}_t)\right)$$

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

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

Optimization on the Sphere

- Training details:
  - Incorporate angular distances into Riemannian optimization



  - Multiply the Euclidean gradient with its angular distance from the current point

$$\boldsymbol{x}_{t+1} = R_{\boldsymbol{x}_t}\left(-\eta_t \left(1 + \frac{\boldsymbol{x}_t^\top \nabla f(\boldsymbol{x}_t)}{\|\nabla f(\boldsymbol{x}_t)\|}\right)\left(I - \boldsymbol{x}_t \boldsymbol{x}_t^\top\right) \nabla f(\boldsymbol{x}_t)\right).$$

Experiments

- Word similarity results:

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 pre-training tasks like predicting masked words and sentence relationships, which have no direct relation to word similarity

Experiments

- Document clustering results:

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

Experiments

- Training efficiency:

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**

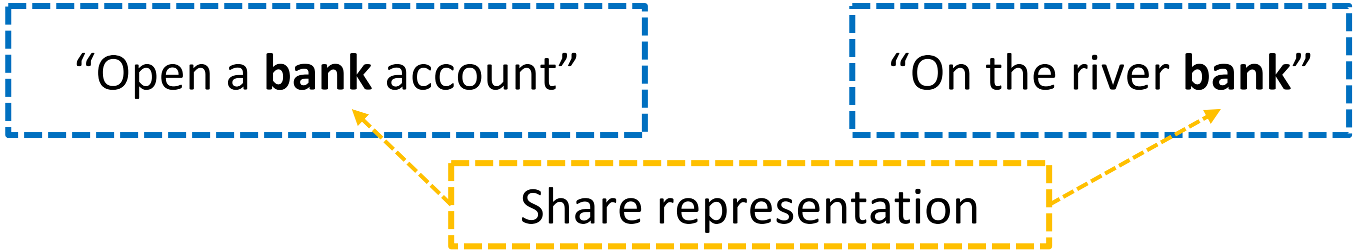
- Why is JoSE training efficient?
  - Other models’ objectives contain many non-linear operations (Word2Vec & fastText's objectives involve exponential functions; GloVe's objective involves logarithm functions), while JoSE only has linear terms in the objective

Outline

- Introduction to text embeddings
- Local context-based word embeddings
- Joint local and global context-based text embeddings
- Deep contextualized embeddings via neural language models
- Extend unsupervised embeddings to incorporate weak supervision

From Context-Free Embedding to Contextualized Embedding

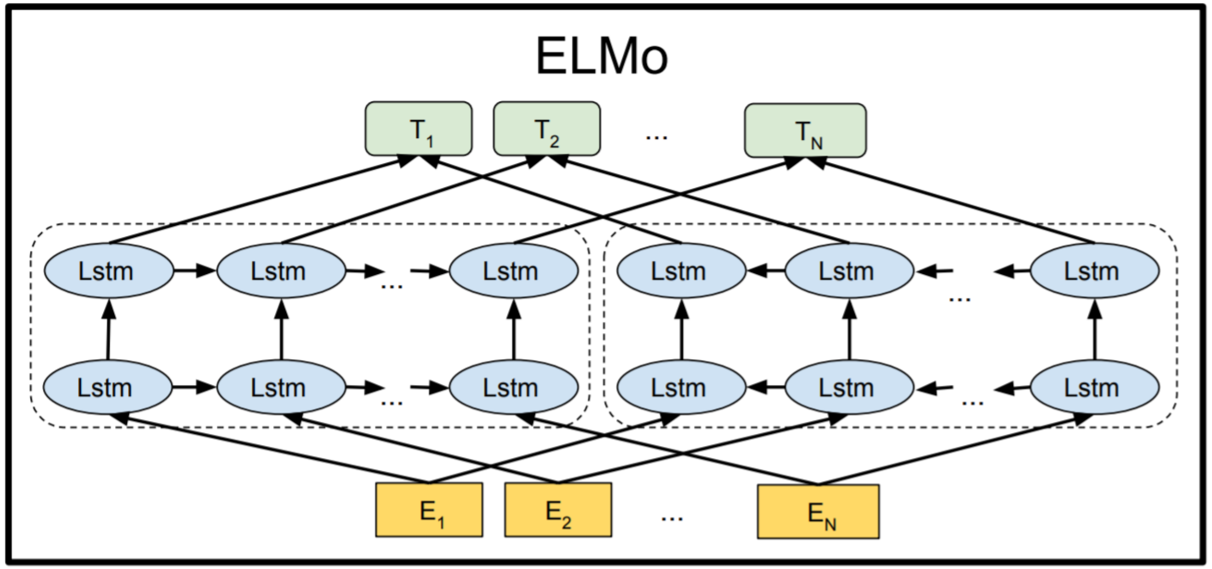
- Previous unsupervised word embeddings like Word2Vec and GloVe learn **context-free** word embedding
  - Each word has one representation regardless of specific contexts it appears in
  - E.g. "bank" is a polysemy, but only has one representation



- Deep neural language models overcome this problem by learning **contextualized** word semantics

ELMo: Deep contextualized word representations

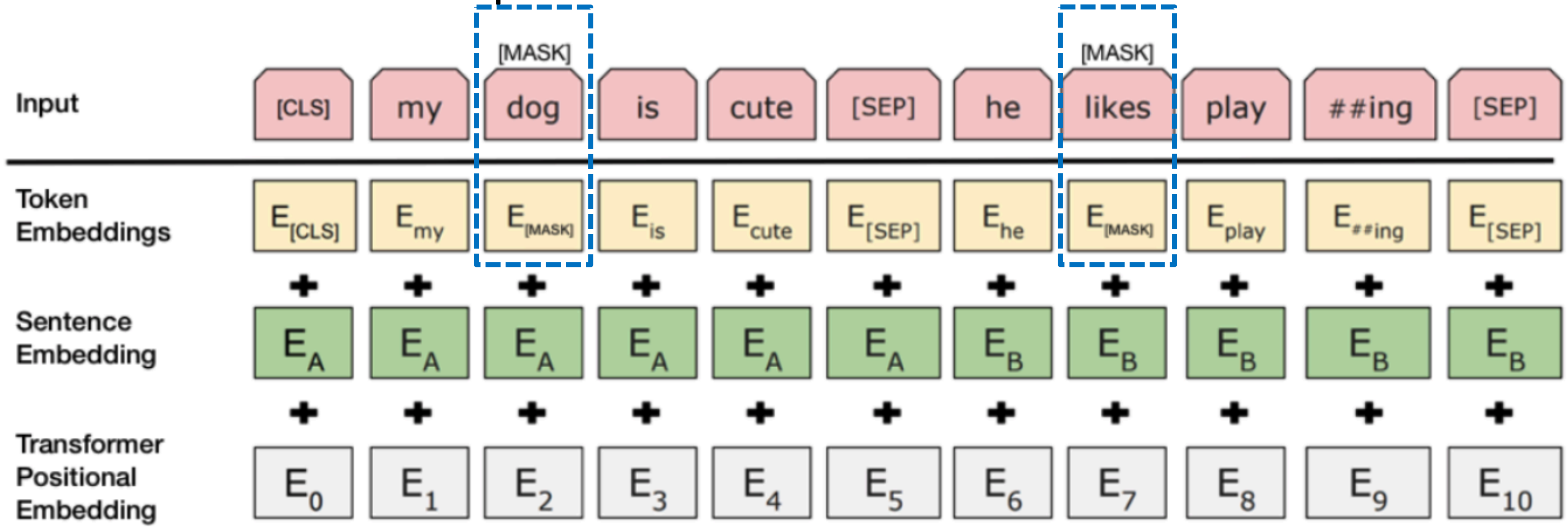
- Word representations are learned functions of the internal states of a deep bi-directional LSTMs
- Results in a pre-trained network that benefits several downstream tasks (e.g. Sentiment analysis, Named entity extraction, Question answering)
- However, left-to-right and right-to-left LSTMs are **independently**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

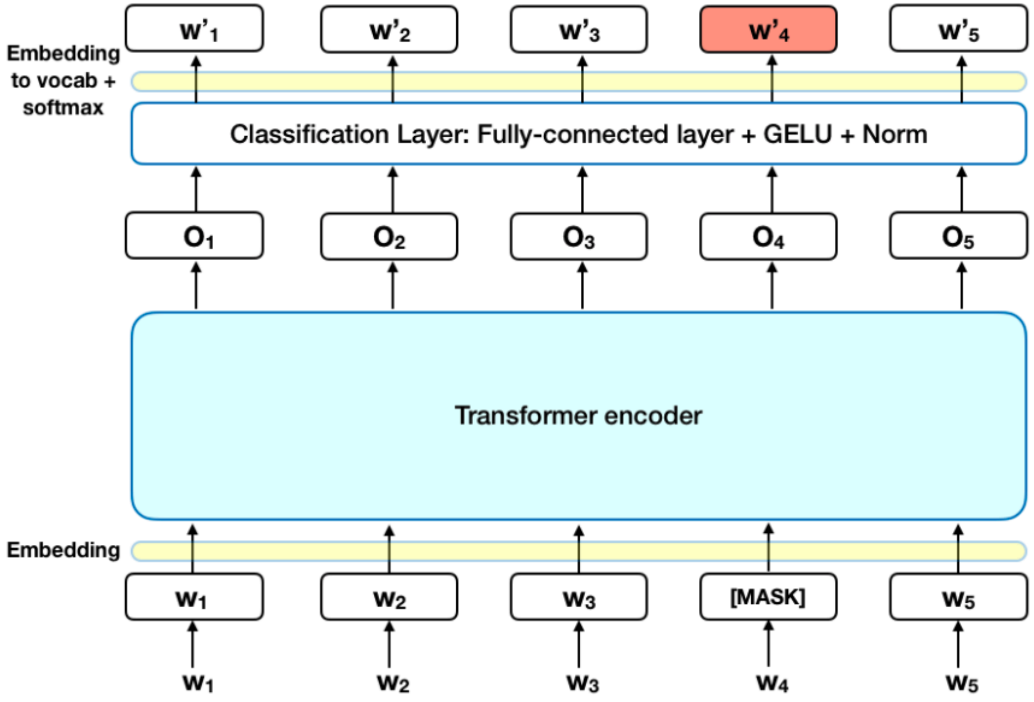
- Bidirectional: BERT leverages a Masked LM learning to introduce **real bidirectionality** training
- Masked LM: With 15% words randomly masked, the model learns bidirectional contextual information to predict the masked words



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

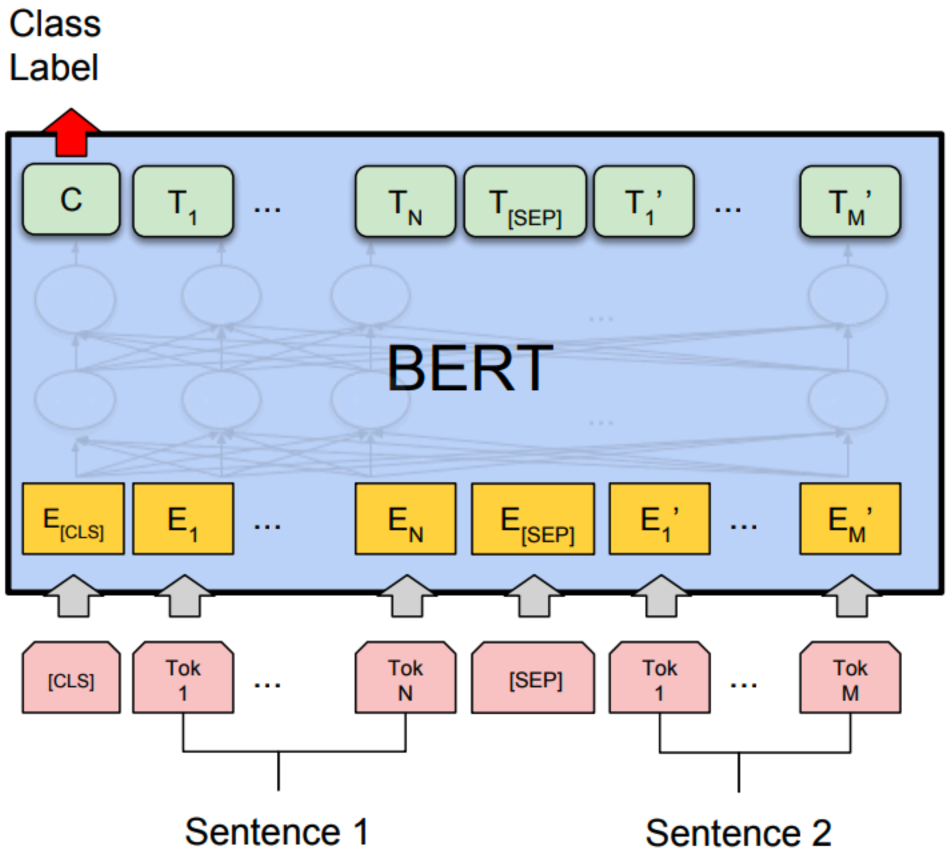
BERT: Deep Bidirectional Transformers

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



BERT: Next Sentence Prediction

- Next Sentence Prediction: learn to predict if the second sentence in the pair is the subsequent sentence in the original document



RoBERTa

- Several simple modifications that make BERT more **effective**:
  - train the model longer, with bigger batches over more data
  - remove the next sentence prediction objective
  - train on longer sequences
  - dynamically change the masking pattern applied to the training data

Model	**data**	**bsz**	**steps**	**SQuAD** (v1.1/2.0)	**MNLI-m**	**SST-2**
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

ALBERT

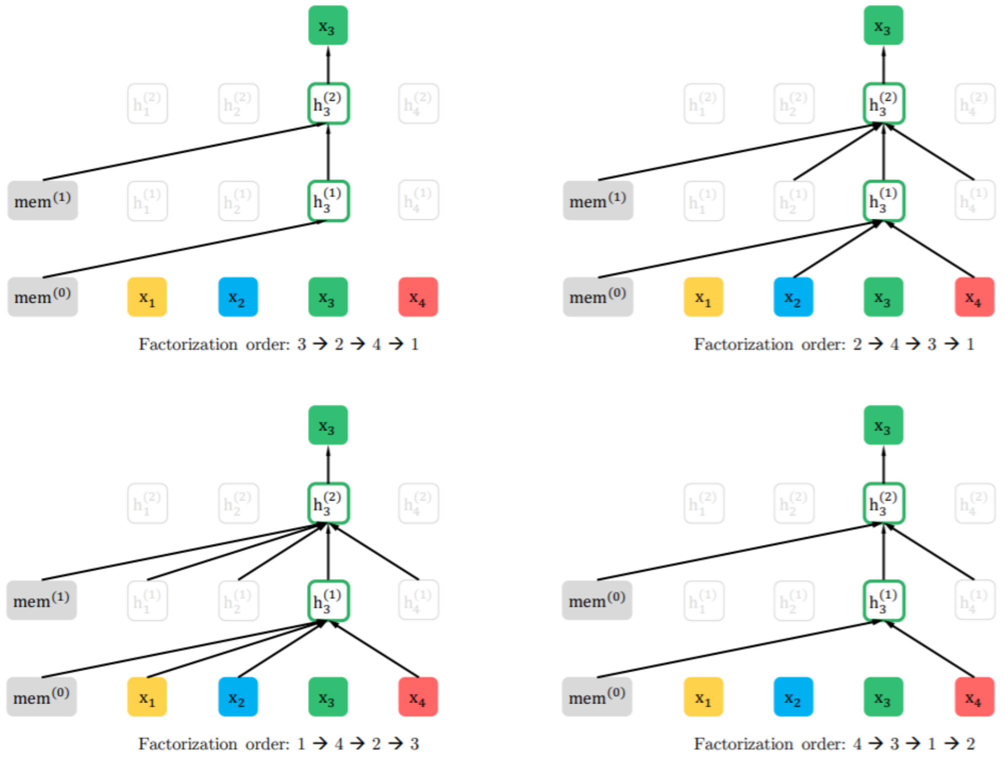
- Simple modifications that make BERT more **efficient**:
  - Factorized embedding parameterization: use lower-dimensional token embeddings; project token embeddings to hidden layer dimension
  - Cross-layer parameter sharing: share feed-forward network parameters/attention parameters across layers
  - Inter-sentence coherence loss: change the next sentence prediction task to sentence order prediction

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. ICLR.

XLNet: Autoregressive Language Modeling

- Issues with BERT: Masked tokens are predicted independently, and [MASK] token brings discrepancy between pre-training and fine-tuning
- XLNet uses Permutation Language Modeling



- 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

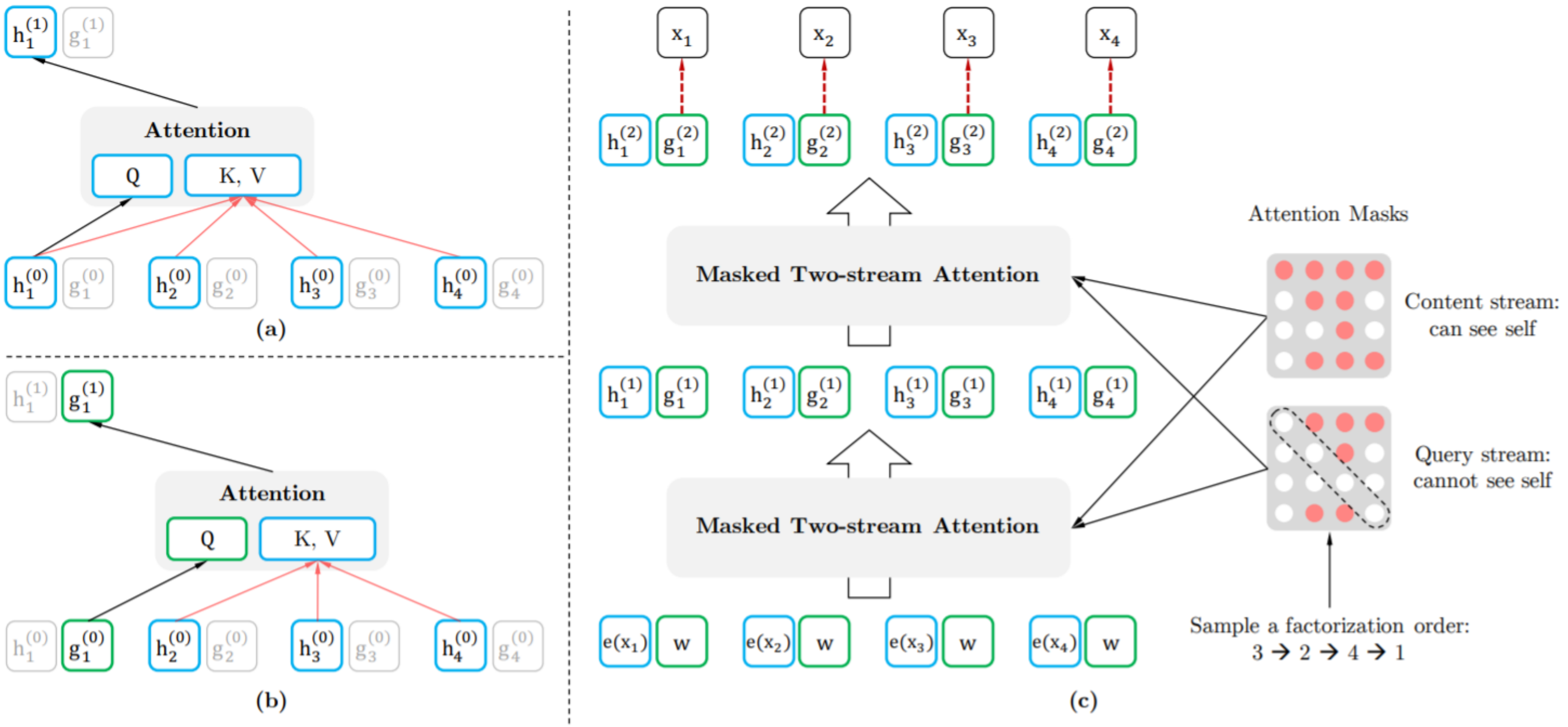
Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv:1906.08237.

XLNet: Two-Stream Self-Attention

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

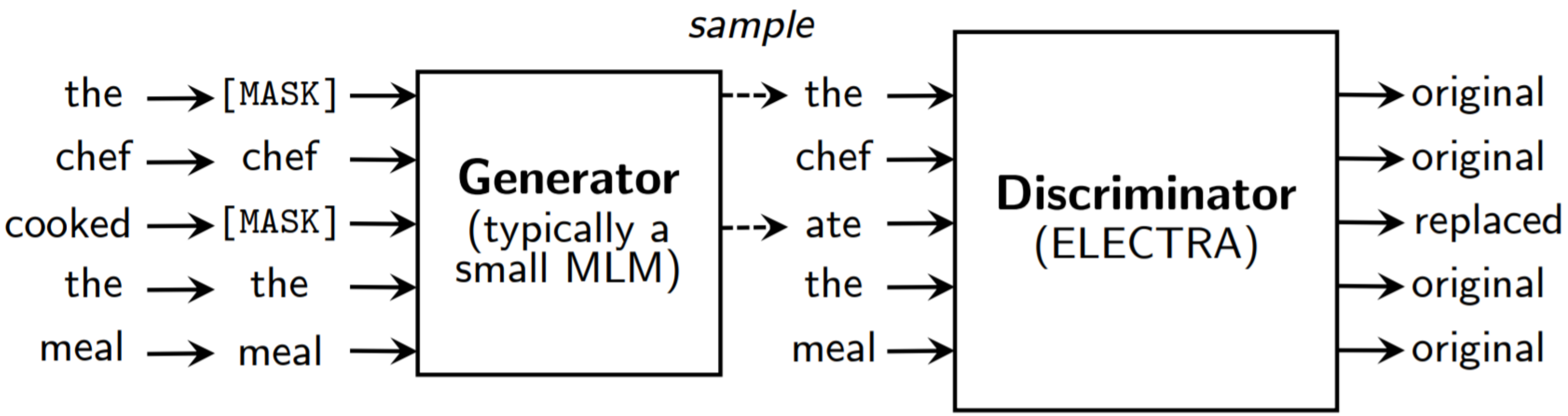
Content representation

Query representation



ELECTRA

- Change masked language modeling to a more sample-efficient pre-training task, **replaced token detection**
- Why more efficient:
  - Replaced token detection trains on all tokens, instead of just on those that are masked (15%)
  - The generator trained with MLM is small (parameter size is ~1/10 of discriminator)
  - The discriminator is trained with a binary classification task, instead of MLM (classification over the entire vocabulary)



Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. ICLR.

ELECTRA

- State-of-the-art GLUE (General Language Understanding Evaluation) test performance with the same compute (measured by Floating Point Operations)

Model	Train FLOPs	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	WNLI	Avg.*	Score
BERT	1.9e20 (0.06x)	60.5	94.9	85.4	86.5	89.3	86.7	92.7	70.1	65.1	79.8	80.5
RoBERTa	3.2e21 (1.02x)	67.8	96.7	89.8	91.9	90.2	90.8	95.4	88.2	89.0	88.1	88.1
ALBERT	3.1e22 (10x)	69.1	**97.1**	**91.2**	92.0	90.5	**91.3**	–	89.2	91.8	89.0	–
XLNet	3.9e21 (1.26x)	70.2	**97.1**	90.5	**92.6**	90.4	90.9	–	88.5	**92.5**	89.1	–
ELECTRA	3.1e21 (1x)	**71.7**	**97.1**	90.7	92.5	**90.8**	**91.3**	**95.8**	**89.8**	**92.5**	**89.5**	**89.4**

Outline

- ❑ Introduction to text embeddings
- ❑ Local context-based word embeddings
- ❑ Joint local and global context-based text embeddings
- ❑ Deep contextualized embeddings via neural language models
- ❑ Extend unsupervised embeddings to incorporate weak supervision

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 will introduce a weakly-supervised text embedding method in Part 3

good	bad
decent	*good* (×)
great	terrible
tasty	poor
yummy	horrible
bad (×)	awful
alright	*alright* (×)
fantastic	weird
impressive	frustrating
weak (×)	harsh
disappointing (×)	*decent* (×)

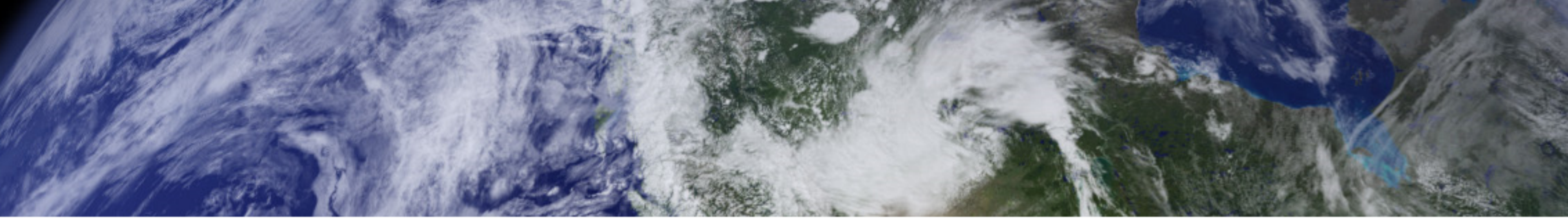
Unsupervised word embedding (Word2Vec) fails to discriminate opposite meaning words

References

- Abu-El-Haija, S., Perozzi, B., Al-Rfou', R., & Alemi, A.A. (2018). Watch Your Step: Learning Node Embeddings via Graph Attention. NeurIPS.
- 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.
- Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. ICLR.
- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. ICLR.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. NIPS.

References (Continued)

- Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. CoRR, abs/1301.3781.
- Meng, Y., Huang, J., Wang, G., Zhang, C., Zhuang, H., Kaplan, L.M., & Han, J. (2019). Spherical Text Embedding. NeurIPS.
- Nickel, M., & Kiela, D. (2017). Poincaré Embeddings for Learning Hierarchical Representations. NIPS.
- Nickel, M., & Kiela, D. (2018). Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. ICML.
- Pennington, J., Socher, R., & Manning, C.D. (2014). Glove: Global Vectors for Word Representation. EMNLP.
- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M.P., Clark, C., Lee, K., & Zettlemoyer, L.S. (2018). Deep contextualized word representations. NAACL.
- Tifrea, A., Bécigneul, G., & Ganea, O. (2019). Poincare Glove: Hyperbolic Word Embeddings. ICLR.
- Turian, J.P., Ratinov, L., & Bengio, Y. (2010). Word Representations: A Simple and General Method for Semi-Supervised Learning. ACL.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. NeurIPS.



Q&A