Modeling Topics and Knowledge Bases with Embeddings

Dat Quoc Nguyen

Department of Computing
Macquarie University
Sydney, Australia

August 2016

Joint work with

- Mark Johnson
- Kairit Sirts
- Lizhen Qu
- Richard Billingsley
- Lan Du

Outline

1. General introduction

2. Improving topic models with word embeddings
   - Introduction
   - Latent-feature topic models
   - Experimental evaluation
   - Summary

3. A new embedding model for knowledge base completion
   - Introduction
   - Our neighborhood mixture model
   - Experimental evaluation
   - Summary

General introduction

Use vector representations for improving topic models as well as for improving link prediction in knowledge bases (i.e. knowledge base completion)

- Incorporate word embeddings trained on large external corpora to improve topic modeling on smaller datasets

  Nguyen et al. "Improving Topic Models with Latent Feature Word Representations." *Transactions of ACL*, 2015, vol. 3, pp. 299-313.

- Predict the missing relationships between entities in knowledge bases

  Nguyen et al. "Neighborhood Mixture Model for Knowledge Base Completion." In *Proceedings of CoNLL 2016*, pp. 40-50.

  Nguyen et al. "STransE: a novel embedding model of entities and relationships in knowledge bases." In *Proceedings of NAACL-HLT 2016*, pp. 460-466.

Outline

Outline

Introduction

- *Topic models* take a corpus of documents as input, and
  - Learn a set of latent *topics* for the corpus
  - Infer *document-to-topic* and *topic-to-word* distributions from co-occurrence of words within documents
- If the corpus is small and/or the documents are short, the topics will be noisy due to the limited information of word co-occurrence
- *Latent word representations* learnt from large external corpora capture various aspects of word meanings
  - We used the pre-trained Word2Vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) word representations

High-level idea

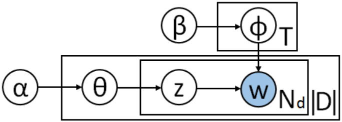
- Use the word representations learnt on a large external corpus to improve the topic-word distributions in a topic model
  - Combine Latent Dirichlet Allocation (Blei et al., 2003) and Dirichlet Multinomial Mixture (Nigam et al., 2000) with the word representations
  - Improvement is greatest on small corpora with short documents

LDA and DMM

- Latent Dirichlet Allocation (LDA)

$\boldsymbol{\theta}_d \sim \text{Dir}(\alpha)$ $\quad$ $z_{d_i} \sim \text{Cat}(\boldsymbol{\theta}_d)$

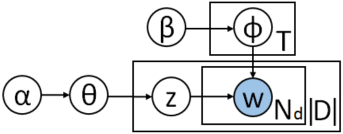
$\phi_z \sim \text{Dir}(\beta)$ $\quad$ $w_{d_i} \sim \text{Cat}(\phi_{z_{d_i}})$



- Dirichlet Multinomial Mixture (DMM) model: one-topic-per-document

$\boldsymbol{\theta} \sim \text{Dir}(\alpha)$ $\quad$ $z_d \sim \text{Cat}(\boldsymbol{\theta})$

$\phi_z \sim \text{Dir}(\beta)$ $\quad$ $w_{d_i} \sim \text{Cat}(\phi_{z_d})$



- Inference is typically performed with a *Gibbs sampler*, integrating out $\boldsymbol{\theta}$ and ϕ (Griffiths et al., 2004; Yin and Wang, 2014)

Outline

1. General introduction
2. Improving topic models with word embeddings
   - Introduction
   - Latent-feature topic models
   - Experimental evaluation
   - Summary
3. A new embedding model for knowledge base completion

Latent-feature topic-to-word distributions

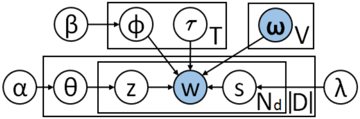
- We assume that each word w is associated with a *word vector* $\boldsymbol{\omega}_w$
- We learn a *topic vector* $\boldsymbol{\tau}_t$ for each topic t
- We use these to define a latent feature topic-to-word distribution $\mathrm{CatE}(w)$ over words:

$$\mathrm{CatE}(w \mid \boldsymbol{\tau}_t \boldsymbol{\omega}^\top) \quad \propto \quad \exp(\boldsymbol{\tau}_t \cdot \boldsymbol{\omega}_w)$$

  - $\boldsymbol{\tau}_t \boldsymbol{\omega}^\top$ is a vector of unnormalized scores, one per word
- In our topic models, we *mix the* CatE *distribution* with a multinomial distribution over words
  - Combine information from a large, general corpus (via the CatE distribution) and a smaller but more specific corpus (via the multinomial distribution)
  - Use a Boolean *indicator variable* that records whether a word is generated from CatE or the multinomial distribution

The Latent Feature LDA (LF-LDA) model

$$\boldsymbol{\theta}_d \sim \text{Dir}(\alpha) \qquad z_{d_i} \sim \text{Cat}(\boldsymbol{\theta}_d)$$
$$\boldsymbol{\phi}_z \sim \text{Dir}(\beta) \qquad s_{d_i} \sim \text{Ber}(\lambda)$$
$$w_{d_i} \sim (1 - s_{d_i})\text{Cat}(\boldsymbol{\phi}_{z_{d_i}}) + s_{d_i}\text{CatE}(\boldsymbol{\tau}_{z_{d_i}}\,\boldsymbol{\omega}^\top)$$



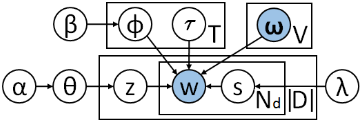
- Replace the topic-to-word Dirichlet multinomial component in LDA with a two-component mixture of a topic-to-word Dirichlet multinomial component and a latent feature topic-to-word component
- s_{d_i} is the Boolean indicator variable indicating whether word w_{d_i} is generated from the latent feature component
- λ is a user-specified hyper-parameter determining how often words are generated from the latent feature component
  - If we estimated λ from data, we expect it would never generate through the latent feature component

The Latent Feature DMM (LF-DMM) model

$\boldsymbol{\theta} \sim \mathrm{Dir}(\alpha)$ $\quad z_d \sim \mathrm{Cat}(\boldsymbol{\theta})$

$\boldsymbol{\phi}_z \sim \mathrm{Dir}(\beta)$ $\quad s_{d_i} \sim \mathrm{Ber}(\lambda)$

$w_{d_i} \sim (1 - s_{d_i})\mathrm{Cat}(\boldsymbol{\phi}_{z_d}) + s_{d_i}\mathrm{CatE}(\boldsymbol{\tau}_{z_d}\,\boldsymbol{\omega}^\top)$



- Replace the topic-to-word Dirichlet multinomial component in DMM with a two-component mixture of a topic-to-word Dirichlet multinomial component and a latent feature topic-to-word component
- s_{d_i} is the Boolean indicator variable indicating whether word w_{d_i} is generated from the latent feature component
- λ is a user-specified hyper-parameter determining how often words are generated from the latent feature component

Inference for the LF-LDA model

- We integrate out $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ as in the Griffiths et al. (2004) sampler, and *interleave MAP estimation for $\boldsymbol{\tau}$ with Gibbs sweeps for the other variables*
- Algorithm outline:

  initialize the word-topic variables z_{d_i} using the LDA sampler
  repeat:

      for each topic t:

        use LBFGS to optimize the L2-regularized log-loss

        $\boldsymbol{\tau}_t = \arg\max_{\boldsymbol{\tau}_t} \mathrm{P}(\boldsymbol{\tau}_t \mid \boldsymbol{z}, \boldsymbol{s})$

      for each document d and each word location i:

        sample z_{d_i} from $\mathrm{P}(z_{d_i} \mid \boldsymbol{z}_{\neg d_i}, \boldsymbol{s}_{\neg d_i}, \boldsymbol{\tau})$

        sample s_{d_i} from $\mathrm{P}(s_{d_i} \mid \boldsymbol{z}, \boldsymbol{s}_{\neg d_i}, \boldsymbol{\tau})$

Inference for the LF-DMM model

- We integrate out $\boldsymbol{\theta}$ and ϕ as in the Yin and Wang (2014) sampler, and *interleave MAP estimation for $\boldsymbol{\tau}$ with Gibbs sweeps*
- Algorithm outline:

  initialize the word-topic variables z_{d_i} using the DMM sampler
  repeat:
  - for each topic t:
    - use LBFGS to optimize the L2-regularized log-loss
      $\boldsymbol{\tau}_t = \arg\max_{\boldsymbol{\tau}_t} \mathrm{P}(\boldsymbol{\tau}_t \mid \boldsymbol{z}, \boldsymbol{s})$
  - for each document d:
    - sample z_d and $\boldsymbol{s}_d$ from $\mathrm{P}(z_d, \boldsymbol{s}_d \mid \boldsymbol{z}_{\neg d}, \boldsymbol{s}_{\neg d}, \boldsymbol{\tau})$

Outline

1. General introduction
2. Improving topic models with word embeddings
   - Introduction
   - Latent-feature topic models
   - Experimental evaluation
   - Summary
3. A new embedding model for knowledge base completion

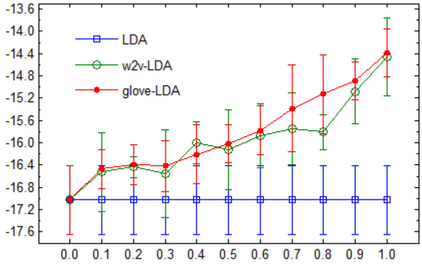
Goals of evaluation

- A topic model learns document-topic and topic-word distributions:
  - *Topic coherence* evaluates the topic-word distributions
  - *Document clustering* and *document classification* evaluate the document-topic distribution
- Do the Word2Vec and Glove word vectors behave differently in topic modelling? (w2v-LDA, glove-LDA, w2v-DMM, glove-DMM)
- We expect that the latent feature component will have *the greatest impact on small corpora*, so our evaluation focuses on them:

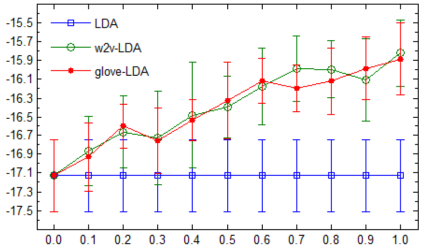
Dataset		# labels	# docs	words/doc	# types
N20	20 newsgroups	20	18,820	103.3	19,572
N20short	$\leq$ 20 words	20	1,794	13.6	6,377
N20small	400 docs	20	400	88.0	8,157
TMN	TagMyNews	7	32,597	18.3	13,428
TMNtitle	TagMyNews titles	7	32,503	4.9	6,347
Twitter		4	2,520	5.0	1,390

Topic coherence evaluation

- Lau et al. (2014) showed that *human scores on a word intrusion task* are highly correlated with the *normalized pointwise mutual information* (NPMI)
- We found latent feature vectors produced a *significant improvement of NPMI scores on all models and corpora*
    - Greatest improvement when $\lambda = 1$ (unsurprisingly)



20 topics



40 topics

NPMI scores on the N20short dataset, varying the mixture weight λ from 0.0 to 1.0.

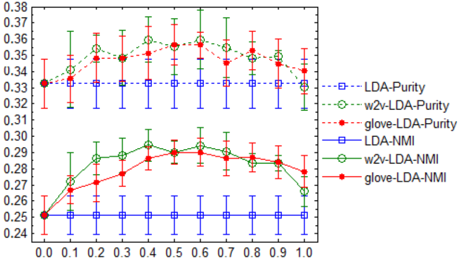
w2v-DMM on TagMyNews titles corpus

Topic 1		Topic 3		Topic 4	
DMM	w2v-DMM	DMM	w2v-DMM	DMM	w2v-DMM
japan	japan	u.s.	prices	egypt	libya
nuclear	nuclear	oil	sales	china	egypt
u.s.	u.s.	japan	oil	u.s	**iran**
crisis	plant	prices	u.s.	mubarak	**mideast**
plant	**quake**	stocks	profit	bin	**opposition**
china	radiation	sales	stocks	libya	**protests**
libya	**earthquake**	profit	japan	laden	**leader**
radiation	**tsunami**	fed	rise	france	**syria**
u.n.	**nuke**	rise	**gas**	bahrain	u.n.
vote	crisis	growth	growth	air	**tunisia**
korea	**disaster**	wall	shares	report	**chief**
europe	**power**	street	**price**	rights	**protesters**
government	**oil**	china	**profits**	court	mubarak
election	**japanese**	fall	**rises**	u.n.	**crackdown**
deal	**plants**	shares	**earnings**	war	bahrain

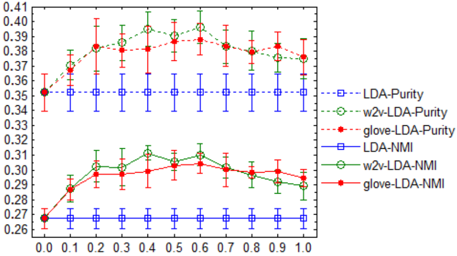
- Table shows the 15 most probable topical words found by 20-topic w2v-DMM on the TMNtitle corpus
- Words found by DMM but not by w2v-DMM are underlined
- Words found by w2v-DMM but not DMM are in bold

Document clustering evaluation (1)

- Cluster documents by assigning them to the *highest probability topic*
- Evaluate clusterings by *purity* and *normalized mutual information* (NMI)



20 topics



40 topics

Purity and NMI results on the N20short dataset, varying the mixture weight λ from 0.0 to 1.0.

- In general, best results with $\lambda = 0.6$

$\Rightarrow$ Set $\lambda = 0.6$ in all further experiments

Document clustering evaluation (2)

Data	Method	Purity		NMI	
		T=4	T=20	T=4	T=20
Twitter	LDA	0.559 ± 0.020	0.614 ± 0.016	0.196 ± 0.018	0.174 ± 0.008
	w2v-LDA	**0.598** ± 0.023	**0.635** ± 0.016	**0.249** ± 0.021	**0.191** ± 0.011
	glove-LDA	0.597 ± 0.016	**0.635** ± 0.014	0.242 ± 0.013	**0.191** ± 0.007
	Improve.	**0.039**	**0.021**	**0.053**	**0.017**
Twitter	DMM	0.523 ± 0.011	0.619 ± 0.015	0.222 ± 0.013	0.213 ± 0.011
	w2v-DMM	**0.589** ± 0.017	0.655 ± 0.015	0.243 ± 0.014	0.215 ± 0.009
	glove-DMM	0.583 ± 0.023	**0.661** ± 0.019	**0.250** ± 0.020	**0.223** ± 0.014
	Improve.	**0.066**	**0.042**	**0.028**	**0.01**

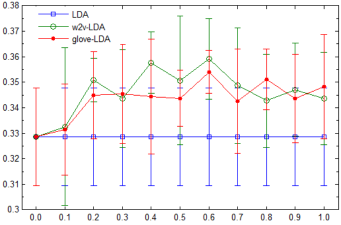
- On the short, our models obtain better clustering results than the baseline models:
  - on N20small, we get 6.0% improvement on NMI at $T = 6$
  - on TMN and TMNtitle, we obtain 6.1% and 2.5% higher Purity at $T = 80$

Document clustering evaluation (3)

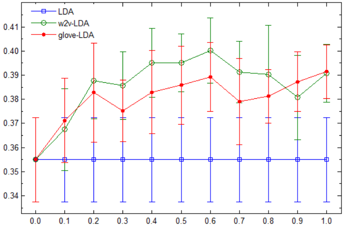
- For small $T \leq 7$, on the large datasets of N20, TMN and TMNtitle, our models and baseline models obtain similar clustering results
- With larger T, our models perform better than baselines on the short TMN and TMNtitle datasets. On the N20 dataset, the baseline LDA model obtains slightly higher clustering results than ours
- No reliable difference between Word2Vec and Glove vectors

Document classification (1)

- Use SVM to predict the ground truth label from the topic-proportion vector of each document



20 topics



40 topics

F_1 scores on N20short dataset, varying the mixture weight λ from 0.0 to 1.0.

Data	Method	$\lambda = 0.6$			
		T=6	T=20	T=40	T=80
N20small	LDA	0.204 ± 0.020	0.392 ± 0.029	0.459 ± 0.030	0.477 ± 0.025
	w2v-LDA	**0.213** ± 0.018	**0.442** ± 0.025	**0.502** ± 0.031	**0.509** ± 0.022
	glove-LDA	0.181 ± 0.011	0.420 ± 0.025	0.474 ± 0.029	0.498 ± 0.012
	Improve.	**0.009**	**0.05**	**0.043**	**0.032**

Document classification (2)

Data	Method	$\lambda = 0.6$			
		T=7	T=20	T=40	T=80
TMN	DMM	0.607 $\pm$ 0.040	0.694 $\pm$ 0.026	0.712 $\pm$ 0.014	0.721 $\pm$ 0.008
	w2v-DMM	0.607 $\pm$ 0.019	0.736 $\pm$ 0.025	**0.760** $\pm$ 0.011	0.771 $\pm$ 0.005
	glove-DMM	**0.621** $\pm$ 0.042	**0.750** $\pm$ 0.011	0.759 $\pm$ 0.006	**0.775** $\pm$ 0.006
	Improve.	**0.014**	**0.056**	**0.048**	**0.054**
TMNtitle	DMM	0.500 $\pm$ 0.021	0.600 $\pm$ 0.015	0.630 $\pm$ 0.016	0.652 $\pm$ 0.005
	w2v-DMM	0.528 $\pm$ 0.028	0.663 $\pm$ 0.008	0.682 $\pm$ 0.008	**0.681** $\pm$ 0.006
	glove-DMM	**0.565** $\pm$ 0.022	**0.680** $\pm$ 0.011	**0.684** $\pm$ 0.009	**0.681** $\pm$ 0.004
	Improve.	**0.065**	**0.08**	**0.054**	**0.029**

Data	Method	$\lambda = 0.6$			
		T=4	T=20	T=40	T=80
Twitter	LDA	0.526 $\pm$ 0.021	0.636 $\pm$ 0.011	0.650 $\pm$ 0.014	0.653 $\pm$ 0.008
	w2v-LDA	**0.578** $\pm$ 0.047	0.651 $\pm$ 0.015	0.661 $\pm$ 0.011	**0.664** $\pm$ 0.010
	glove-LDA	0.569 $\pm$ 0.037	**0.656** $\pm$ 0.011	**0.662** $\pm$ 0.008	0.662 $\pm$ 0.006
	Improve.	**0.052**	**0.02**	**0.012**	**0.011**
Twitter	DMM	0.469 $\pm$ 0.014	0.600 $\pm$ 0.021	0.645 $\pm$ 0.009	0.665 $\pm$ 0.014
	w2v-DMM	**0.539** $\pm$ 0.016	0.649 $\pm$ 0.016	0.656 $\pm$ 0.007	0.676 $\pm$ 0.012
	glove-DMM	0.536 $\pm$ 0.027	**0.654** $\pm$ 0.019	**0.657** $\pm$ 0.008	**0.680** $\pm$ 0.009
	Improve.	**0.07**	**0.054**	**0.012**	**0.015**

Outline

Conclusions and future directions

- Latent feature vectors induced from large external corpora can be used to improve topic modeling
  - Latent features significantly improve topic coherence across a range of corpora with both the LDA and DMM models
  - Document clustering and document classification also significantly improve, even though these depend directly only on the document-topic distribution
- The improvements were greatest for small document collections and/or for short documents
- We did not detect any reliable difference between Word2Vec and Glove vectors
- Retrain the word vectors to fit the topic-modeling corpus
- More sophisticated latent-feature models of topic-word distributions
- More efficient training procedures

Outline

Outline

- Knowledge bases (KBs) of real-world triple facts (head entity, relation, tail entity) are useful resources for NLP tasks
- **Issue**: large KBs are still far from complete
- So it is useful to perform *link prediction in KBs* or *knowledge base completion* (KBC): predict which triples not in a knowledge base are likely to be true

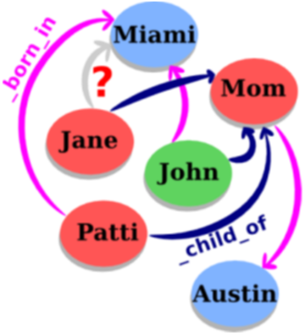


Figure extracted from "Jason Weston and Antoine Bordes. 2014. Embedding Methods for NLP. *EMNLP 2014 tutorial*."

Introduction

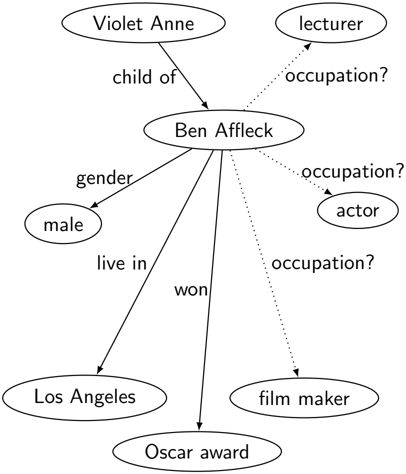
- Embedding models for KBC:
  - Associate entities and/or relations with dense feature vectors or matrices
  - Obtain SOTA performance and generalize to large KBs
- Most embedding models for KBC learn only from triples
- Recent works show that the relation paths between entities in KBs provide useful information and improve KBC

$$
\begin{aligned}
&(\text{Harrison Ford}, \textbf{born\_in\_hospital}/\mathrm{r}_1, \text{Swedish Covenant Hospital})\\
\Rightarrow&(\text{Swedish Covenant Hospital}, \textbf{located\_in\_city}/\mathrm{r}_2, \text{Chicago})\\
\Rightarrow&(\text{Chicago}, \textbf{city\_in\_country}/\mathrm{r}_3, \text{United States})
\end{aligned}
$$

Relation path $\mathrm{p} = \{\mathrm{r}_1, \mathrm{r}_2, \mathrm{r}_3\}$ is useful for predicting the relationship "*nationality*" between the head and tail entities

- **Our motivation**: neighborhoods could provide lots of useful information for predicting the relationship between the entities

$$\begin{aligned}&\text{Ben\_Affleck}\\&= \omega_{r,1}(\text{Violet\_Anne}, \text{child\_of})\\&\quad + \omega_{r,2}(\text{male}, \text{gender}^{-1})\\&\quad + \omega_{r,3}(\text{Los\_Angeles}, \text{live\_in}^{-1})\\&\quad + \omega_{r,4}(\text{Oscar\_award}, \text{won}^{-1})\end{aligned}$$



Outline

1. General introduction
2. Improving topic models with word embeddings
3. A new embedding model for knowledge base completion
   - Introduction
   - Our neighborhood mixture model
   - Experimental evaluation
   - Summary

Our neighbor-based entity representation

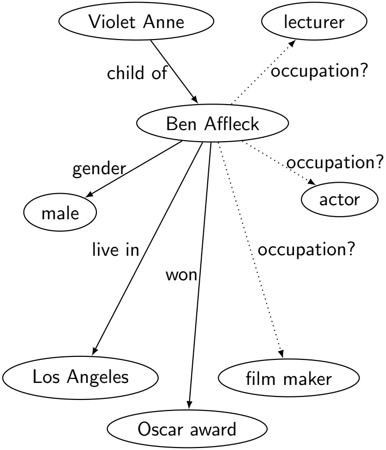
$\mathcal{E} = \{\text{Ben\_Affleck}, \text{Los\_Angeles}, \ldots\}$

$\mathcal{R} = \{\text{live\_in}, \text{won}, \text{child\_of}, \text{gender}, \ldots\}$

$$\begin{aligned}\mathcal{G} = \{&(\text{Violet\_Anne}, \text{child\_of}, \text{Ben\_Affleck}),\\ &(\text{Ben\_Affleck}, \text{won}, \text{Oscar\_award}),\\ &(\text{Ben\_Affleck}, \text{live\_in}, \text{Los\_Angeles}), \ldots\}\end{aligned}$$

$\mathcal{N}_e$ is the set of all entity and relation pairs that are neighbors for entity e

$$\begin{aligned}\mathcal{N}_{\text{Ben\_Affleck}} = \{&(\text{Violet\_Anne}, \text{child\_of}),\\ &(\text{male}, \text{gender}^{-1}),\\ &(\text{Los\_Angeles}, \text{live\_in}^{-1}),\\ &(\text{Oscar\_award}, \text{won}^{-1})\}\end{aligned}$$



Our neighbor-based entity representation

- $\boldsymbol{v}_e \in \mathbb{R}^k$: k-dimensional "base" vector associated with entity e
- $\boldsymbol{u}_{e,r} \in \mathbb{R}^k$: relation-specific entity vector, $e \in \mathcal{E}$, $r \in \mathcal{R} \cup \mathcal{R}^{-1}$
- The neighborhood-based entity representation $\boldsymbol{\vartheta}_{e,r}$ for an entity e for predicting the relation r is defined as follows:

$$\boldsymbol{\vartheta}_{e,r} = a_e \boldsymbol{v}_e + \sum_{(e',r') \in \mathcal{N}_e} b_{r,r'} \boldsymbol{u}_{e',r'} \tag{1}$$

a_e and $b_{r,r'}$ are the mixture weights that are constrained to sum to 1:

$$a_e \propto \delta + \exp \alpha_e \tag{2}$$

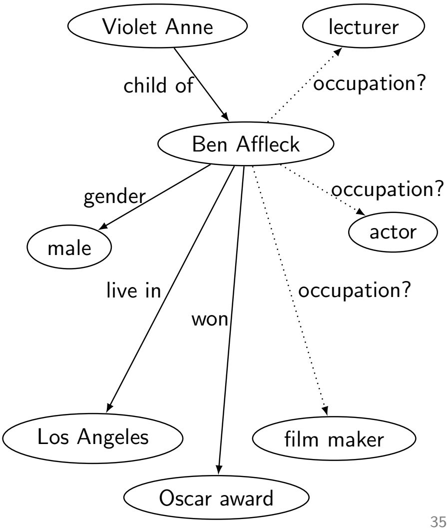
$$b_{r,r'} \propto \exp \beta_{r,r'} \tag{3}$$

$\delta \geqslant 0$: hyper-parameter
α_e, $\beta_{r,r'}$: learnable exponential mixture parameters

Our neighbor-based entity representation

$$\boldsymbol{\vartheta}_{e,r} = a_e \boldsymbol{v}_e + \sum_{(e',r')\in\mathcal{N}_e} b_{r,r'} \boldsymbol{u}_{e',r'}$$

$$\begin{aligned}
e &= \text{Ben_Affleck} \\
r &= \text{occupation} \\
\mathcal{N}_e &= \{(\text{Violet_Anne}, \text{child_of}), \\
&\qquad (\text{male}, \text{gender}^{-1}), \\
&\qquad (\text{Los_Angeles}, \text{live_in}^{-1}), \\
&\qquad (\text{Oscar_award}, \text{won}^{-1})\}
\end{aligned}$$



Our new embedding model TransE-NMM for KBC

- Embedding models define for each triple $(h, r, t) \in \mathcal{G}$, a *score function* $f(h, r, t)$ that measures its implausibility
- **Goal**: choose f such that the score $f(h, r, t)$ of a plausible triple (h, r, t) is smaller than the score $f(h', r', t')$ of an implausible triple (h', r', t').
- Entity e and relation r are represented with vectors $\boldsymbol{v}_e \in \mathbb{R}^k$ and $\boldsymbol{v}_r \in \mathbb{R}^k$

$$f(h, r, t)_{\text{TransE}} = \|\boldsymbol{v}_h + \boldsymbol{v}_r - \boldsymbol{v}_t\|_{\ell_{1/2}}$$

- The score function of our new model TransE-NMM is defined as follows:

$$f(h, r, t) = \|\boldsymbol{\vartheta}_{h,r} + \boldsymbol{v}_r - \boldsymbol{\vartheta}_{t,r^{-1}}\|_{\ell_{1/2}} \tag{4}$$

$$\boldsymbol{\vartheta}_{e,r} = a_e \boldsymbol{v}_e + \sum_{(e',r') \in \mathcal{N}_e} b_{r,r'} \boldsymbol{u}_{e',r'}$$

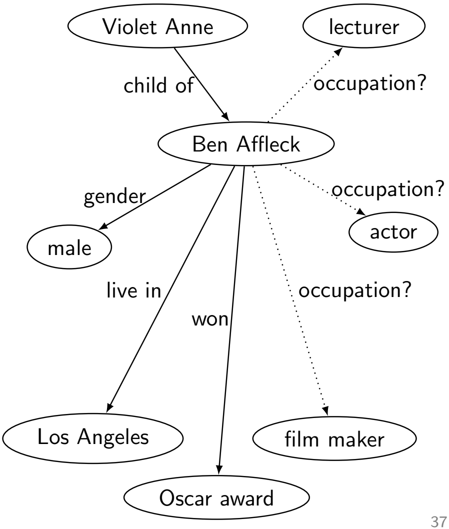
$$\boldsymbol{u}_{e,r} = \boldsymbol{v}_e + \boldsymbol{v}_r \tag{5}$$

$$\boldsymbol{v}_{r^{-1}} = -\boldsymbol{v}_r \tag{6}$$

Our new embedding model TransE-NMM for KBC

$$\boldsymbol{\vartheta}_{e,r} = a_e \boldsymbol{v}_e + \sum_{(e',r') \in \mathcal{N}_e} b_{r,r'} \left(\boldsymbol{v}_{e'} + \boldsymbol{v}_{r'} \right)$$

$$\begin{aligned} e &= \text{Ben\_Affleck} \\ r &= \text{occupation} \\ \mathcal{N}_e &= \{(\text{Violet\_Anne}, \text{child\_of}), \\ &\quad\ (\text{male}, \text{gender}^{-1}), \\ &\quad\ (\text{Los\_Angeles}, \text{live\_in}^{-1}), \\ &\quad\ (\text{Oscar\_award}, \text{won}^{-1})\} \end{aligned}$$



Parameter optimization

- Model parameters:
  - Entity vectors $\boldsymbol{v}_e$
  - Relation type vectors $\boldsymbol{v}_r$
  - $\boldsymbol{\alpha} = \{\alpha_e | e \in \mathcal{E}\}$: entity-specific weights
  - $\boldsymbol{\beta} = \{\beta_{r,r'} | r, r' \in \mathcal{R} \cup \mathcal{R}^{-1}\}$: relation-specific weights
- Minimize the L_2-regularized margin-based objective function:

$$\mathcal{L} = \sum_{\substack{(h,r,t)\in\mathcal{G} \\ (h',r,t')\in\mathcal{G}'_{(h,r,t)}}} [\gamma + f(h,r,t) - f(h',r,t')]_+ + \frac{\lambda}{2}\left(\|\boldsymbol{\alpha}\|_2^2 + \|\boldsymbol{\beta}\|_2^2\right)$$

$$\mathcal{G}'_{(h,r,t)} = \{(h',r,t) \mid h' \in \mathcal{E}, (h',r,t) \notin \mathcal{G}\} \\ \cup \{(h,r,t') \mid t' \in \mathcal{E}, (h,r,t') \notin \mathcal{G}\}$$

  - $[x]_+ = \max(0, x)$
  - γ: the margin hyper-parameter
  - λ: the L_2 regularization parameter
  - Impose constraints during training with RMSProp: $\|\boldsymbol{v}_e\|_2 \leqslant 1$, $\|\boldsymbol{v}_r\|_2 \leqslant 1$

Related work

Model	Score function $f(h, r, t)$
STransE	$\|\mathbf{W}_{r,1}\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_{r,2}\boldsymbol{v}_t\|_{\ell_{1/2}}$; $\mathbf{W}_{r,1}$, $\mathbf{W}_{r,2} \in \mathbb{R}^{k\times k}$; $\boldsymbol{v}_r \in \mathbb{R}^k$
TransE	$\|\boldsymbol{v}_h + \boldsymbol{v}_r - \boldsymbol{v}_t\|_{\ell_{1/2}}$; $\boldsymbol{v}_r \in \mathbb{R}^k$
TransH	$\|(\mathbf{I} - \boldsymbol{r}_p\boldsymbol{r}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} - \boldsymbol{r}_p\boldsymbol{r}_p^\top)\boldsymbol{v}_t\|_{\ell_{1/2}}$ $\boldsymbol{r}_p$, $\boldsymbol{v}_r \in \mathbb{R}^k$; $\mathbf{I}$: Identity matrix size $k \times k$
TransD	$\|(\mathbf{I} + \boldsymbol{r}_p\boldsymbol{h}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} + \boldsymbol{r}_p\boldsymbol{t}_p^\top)\boldsymbol{v}_t\|_{\ell_{1/2}}$ $\boldsymbol{r}_p$, $\boldsymbol{v}_r \in \mathbb{R}^n$; $\boldsymbol{h}_p, \boldsymbol{t}_p \in \mathbb{R}^k$; $\mathbf{I}$: Identity matrix size $n \times k$
TransR	$\|\mathbf{W}_r\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_r\boldsymbol{v}_t\|_{\ell_{1/2}}$; $\mathbf{W}_r \in \mathbb{R}^{n\times k}$; $\boldsymbol{v}_r \in \mathbb{R}^n$
NTN	$\boldsymbol{v}_r^\top tanh(\boldsymbol{v}_h^\top\mathbf{M}_r\boldsymbol{v}_t + \mathbf{W}_{r,1}\boldsymbol{v}_h + \mathbf{W}_{r,2}\boldsymbol{v}_t + \mathbf{b}_r)$ $\boldsymbol{v}_r$, $\mathbf{b}_r \in \mathbb{R}^n$; $\mathbf{M}_r \in \mathbb{R}^{k\times k\times n}$; $\mathbf{W}_{r,1}$, $\mathbf{W}_{r,2} \in \mathbb{R}^{n\times k}$
DISTMULT	$\boldsymbol{v}_h^\top\mathbf{W}_r\boldsymbol{v}_t$; $\mathbf{W}_r$ is a diagonal matrix $\in \mathbb{R}^{k\times k}$
Bilinear-COMP	$\boldsymbol{v}_h^\top\mathbf{W}_{r_1}\mathbf{W}_{r_2}...\mathbf{W}_{r_m}\boldsymbol{v}_t$; $\mathbf{W}_{r_1}, \mathbf{W}_{r_2}, ..., \mathbf{W}_{r_m} \in \mathbb{R}^{k\times k}$
TransE-COMP	$\|\boldsymbol{v}_h + \boldsymbol{v}_{r_1} + \boldsymbol{v}_{r_2} + ... + \boldsymbol{v}_{r_m} - \boldsymbol{v}_t\|_{\ell_{1/2}}$; $\boldsymbol{v}_{r_1}, \boldsymbol{v}_{r_2}, ..., \boldsymbol{v}_{r_m} \in \mathbb{R}^k$
TransE-NMM	$\|\boldsymbol{\vartheta}_{h,r} + \boldsymbol{v}_r - \boldsymbol{\vartheta}_{t,r^{-1}}\|_{\ell_{1/2}}$

Outline

Evaluation: experimental setup

Dataset:	WN11	FB13	NELL186
#R	11	13	186
#E	38,696	75,043	14,463
#Train	112,581	316,232	31,134
#Valid	2,609	5,908	5,000
#Test	10,544	23,733	5,000

- #E: number of entities
- #R: number of relation types
- #Train, #Valid and #Test are the numbers of correct triples in the training, validation, and test sets, respectively
- Each validation and test set also contains the same number of incorrect triples as the number of correct triples

Triple classification task:

- Predict whether a triple (h, r, t) is correct or not
- Set a relation-specific threshold θ_r for each relation type r
- For an unseen test triple (h, r, t), if $f(h, r, t)$ is smaller than θ_r then the triple will be classified as correct, otherwise incorrect
- The relation-specific thresholds are determined by maximizing the micro-averaged accuracy on the validation set

Evaluation: experimental setup

- **Entity prediction task**:
  - Predict h given $(?, r, t)$ or predict t given $(h, r, ?)$ where ? denotes the missing element
  - Corrupt each correct test triple (h, r, t) by replacing either h or t by each of the possible entities in turn
  - Rank these candidates in ascending order of their implausibility value computed by the score function
  - "Raw" and "Filtered" setting protocols in which "Filtered" setting is to filter out before ranking any corrupted triples that appear in the KB
  - Metrics: mean rank (MR), mean reciprocal rank (MRR) and Hits@10 (H10)
- **Relation prediction task**:
  - Predict r given $(h, ?, t)$ where ? denotes the missing element
  - Corrupt each correct test triple (h, r, t) by replacing r by each of the possible relations in turn

Evaluation: quantitative results

Data	Method		Triple class.		Entity prediction			Relation prediction		
			Mic.	Mac.	MR	MRR	H@10	MR	MRR	H@10
WN11	R	TransE	85.21	82.53	4324	0.102	19.21	2.37	0.679	**99.93**
		TransE-NMM	**86.82**	**84.37**	**3466**	**0.123**	**20.59**	**2.14**	**0.687**	99.92
	F	TransE			4304	0.122	21.86	2.37	0.679	**99.93**
		TransE-NMM			**3447**	**0.137**	**23.03**	**2.14**	**0.687**	99.92
FB13	R	TransE	87.57	86.66	9037	0.204	35.39	1.01	0.996	99.99
		TransE-NMM	**88.58**	**87.99**	**8289**	**0.258**	**35.53**	1.01	0.996	**100.0**
	F	TransE			5600	0.213	36.28	1.01	0.996	99.99
		TransE-NMM			**5018**	**0.267**	**36.36**	1.01	0.996	**100.0**
NELL186	R	TransE	92.13	88.96	309	0.192	36.55	8.43	0.580	77.18
		TransE-NMM	**94.57**	**90.95**	**238**	**0.221**	**37.55**	**6.15**	**0.677**	**82.16**
	F	TransE			279	0.268	47.13	8.32	0.602	77.26
		TransE-NMM			**214**	**0.292**	**47.82**	**6.08**	**0.690**	**82.20**

- **Mic.**: Micro-averaged accuracy; **Mac.**: Macro-averaged accuracy
- "R" and "F" denote the "Raw" and "Filtered" settings used in the entity prediction and relation prediction tasks, respectively
- Better results are in **bold**

Evaluation: quantitative results

Method	**W11**	**F13**
TransR	85.9	82.5
CTransR	85.7	-
TransD	<u>86.4</u>	**89.1**
TranSparse-S	<u>86.4</u>	88.2
TranSparse-US	**86.8**	87.5
NTN	70.6	87.2
TransH	78.8	83.3
SLogAn	75.3	85.3
KG2E	85.4	85.3
Bilinear-COMP	77.6	86.1
TransE-COMP	80.3	87.6
TransE	85.2	87.6
TransE-NMM	**86.8**	<u>88.6</u>

Micro-averaged accuracy for triple classification on WN11 and FB13

Results on the NELL186 test set:

Method	**Triple class.**		**Entity pred.**	
	Mic.	Mac.	MR	H@10
TransE-LLE	90.08	84.50	535	20.02
SME-LLE	93.64	89.39	<u>253</u>	37.14
SE-LLE	<u>93.95</u>	88.54	447	31.55
TransE-SkipG	85.33	80.06	385	30.52
SME-SkipG	92.86	<u>89.65</u>	293	**39.70**
SE-SkipG	93.07	87.98	412	31.12
TransE	92.13	88.96	309	36.55
TransE-NMM	**94.57**	**90.95**	**238**	<u>37.55</u>

The entity prediction results are in the "Raw" setting

Evaluation: qualitative results

- Take the relation-specific mixture weights from the learned TransE-NMM
- Extract neighbor relations with the largest mixture weights given a relation

Relation	**Top 3-neighbor relations**
has_instance (WN11)	type_of subordinate_instance_of domain_topic
nationality (FB13)	place_of_birth place_of_death location
CEOof (NELL186)	WorksFor TopMemberOfOrganization PersonLeadsOrganization

Outline

Conclusions and future work

- We introduced a neighborhood mixture model for knowledge base completion by constructing neighbor-based vector representations for entities
- We demonstrated its effect by extending the state-of-the-art embedding model TransE with our neighborhood mixture model
- Our model significantly improves TransE and obtains better results than the other state-of-the-art embedding models on three evaluation tasks
- We plan to apply the neighborhood mixture model to the relation path models to combine the useful information from both relation paths and entity neighborhoods

Thank you for your attention!