Advanced approaches to enhancing word2vec

Yan Song

Outline

- Problems to address
  - Possible enhancement
- Using external semantic resources
- Leanring from language structures
- Leveraging sub word information
- Extensions to other units and tasks

word2vec (or similar models)

- Context-based language modeling
  - Actually, does not do language modeling
- Learning without semantic guidance
  - The only one is frequencies through HS
- Restricted in its implementation
  - Not easy to extend

Problems of word2vec

- Supervision
  - Current word2vec (as well as GloVe) is unsupervised
  - Guidance can be provided to enhance learning

- Possible enhancement
  - Dictionary?
  - Manual annotations?

Problems of word2vec

- Further information integration
  - Lack of high-level structural information
  - Many other different attributes that can help learning

- Possible enhancement
  - Structural knowledge?
  - Sub-word decomposition?

Problems of word2vec

- Flexible Usage
  - Current model does only embedding learning
  - Many possible extensions can be made

- How to?
  - Leverage word embedding results
  - Extend to other task

Retrofitting

- Faruqui et al. (2015)
  - Restrict word relations by semantic lexicons
  - Require initial embeddings
  - Online updating

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

$$q_i = \frac{\sum_{j:(i,j)\in E} \beta_{ij} q_j + \alpha_i \hat{q}_i}{\sum_{j:(i,j)\in E} \beta_{ij} + \alpha_i}$$

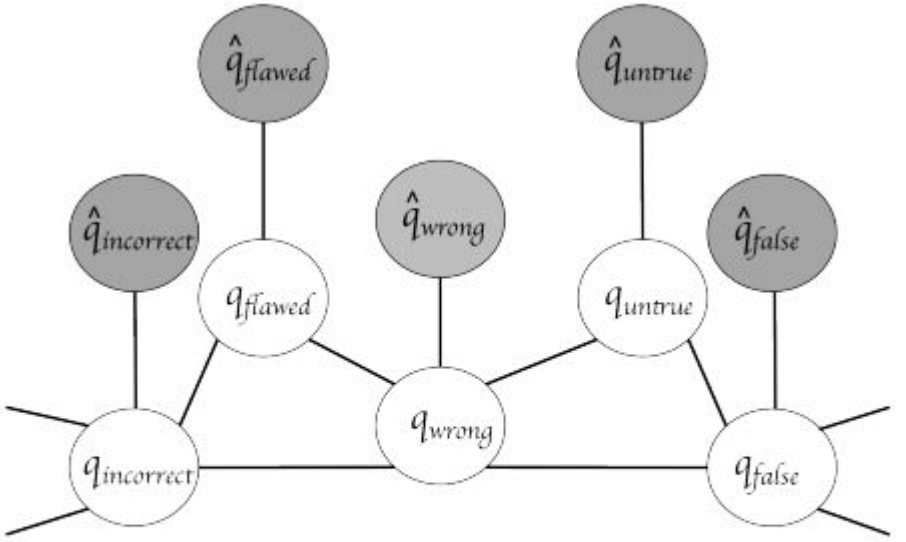


Figure 1: Word graph with edges between related words showing the observed (grey) and the inferred (white) word vector representations.

Joint Learning or Retrofitting

- Kiela et al. (2015)
  - Joint: SG (context) + SG (lexicon)

$$\frac{1}{T}\sum_{t=1}^{T}\left(J_\theta(w_t) + \sum_{w^a \in A_{w_t}} \log p(w^a|w_t)\right)$$

  - Retro: initial embeddings + SG (lexicon)

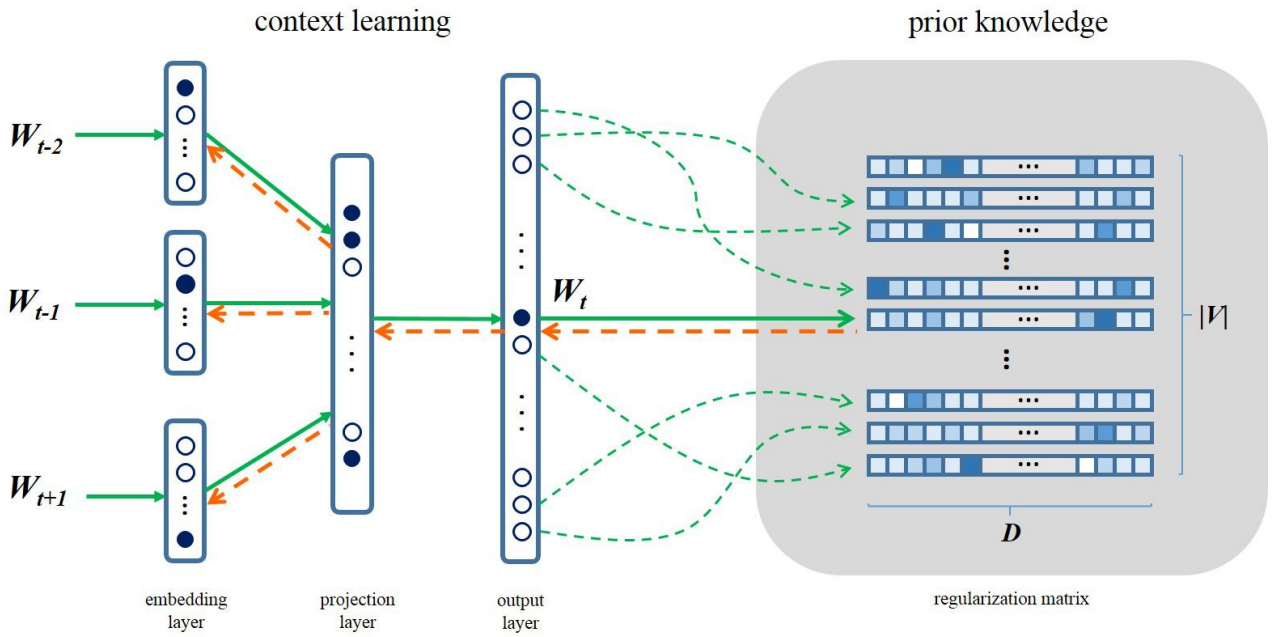
$$\frac{1}{T}\sum_{t=1}^{T}\sum_{w^a \in A_{w_t}} \log p(w^a|w_t)$$

Method	SimLex-999	MEN
Skip-gram	0.31	0.68
Fit-Norms	0.08	0.14
Fit-Thesaurus	0.26	0.14
Joint-Norms-Sampled	0.43	**0.72**
Joint-Norms-All	0.42	0.67
Joint-Thesaurus-Sampled	0.38	0.69
Joint-Thesaurus-All	0.44	0.60
GB-Retrofit-Norms	0.32	0.71
GB-Retrofit-Thesaurus	0.38	0.68
SG-Retrofit-Norms	0.35	0.71
SG-Retrofit-Thesaurus	**0.47**	0.69

Table 1: Spearman ρ on a genuine similarity (SimLex-999) and relatedness (MEN) dataset.

Combination

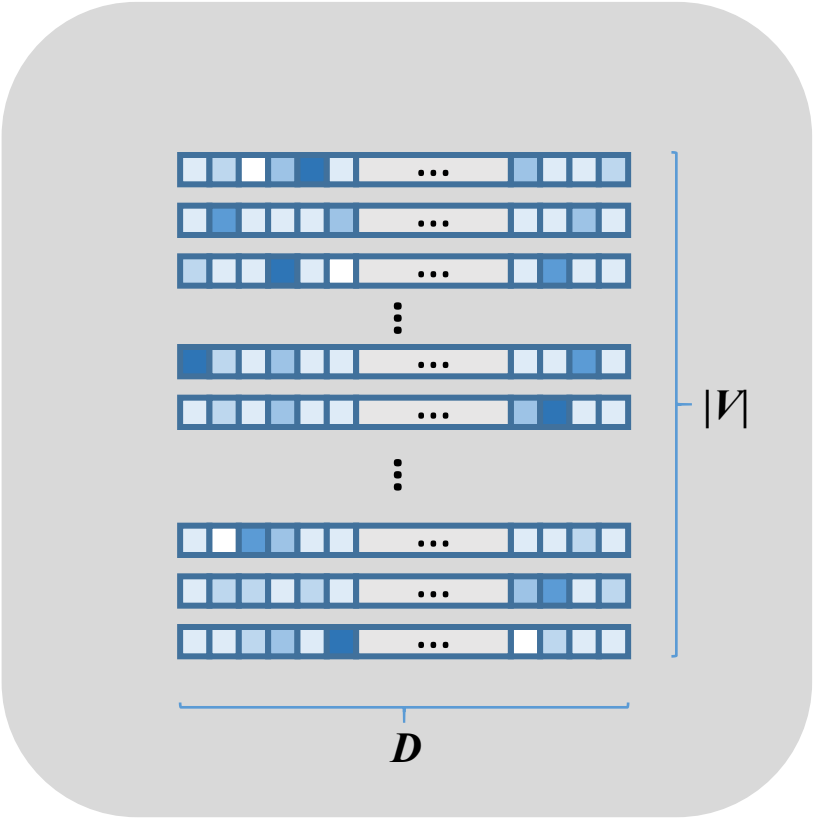
- Song et al. (2017)
  - Context + External knowledge
  - In prediction manner
  - Fit for both CBOW and SG
  - Incorporating
    - Supervised knowledge
    - Unsupervised knowledge



The Idea

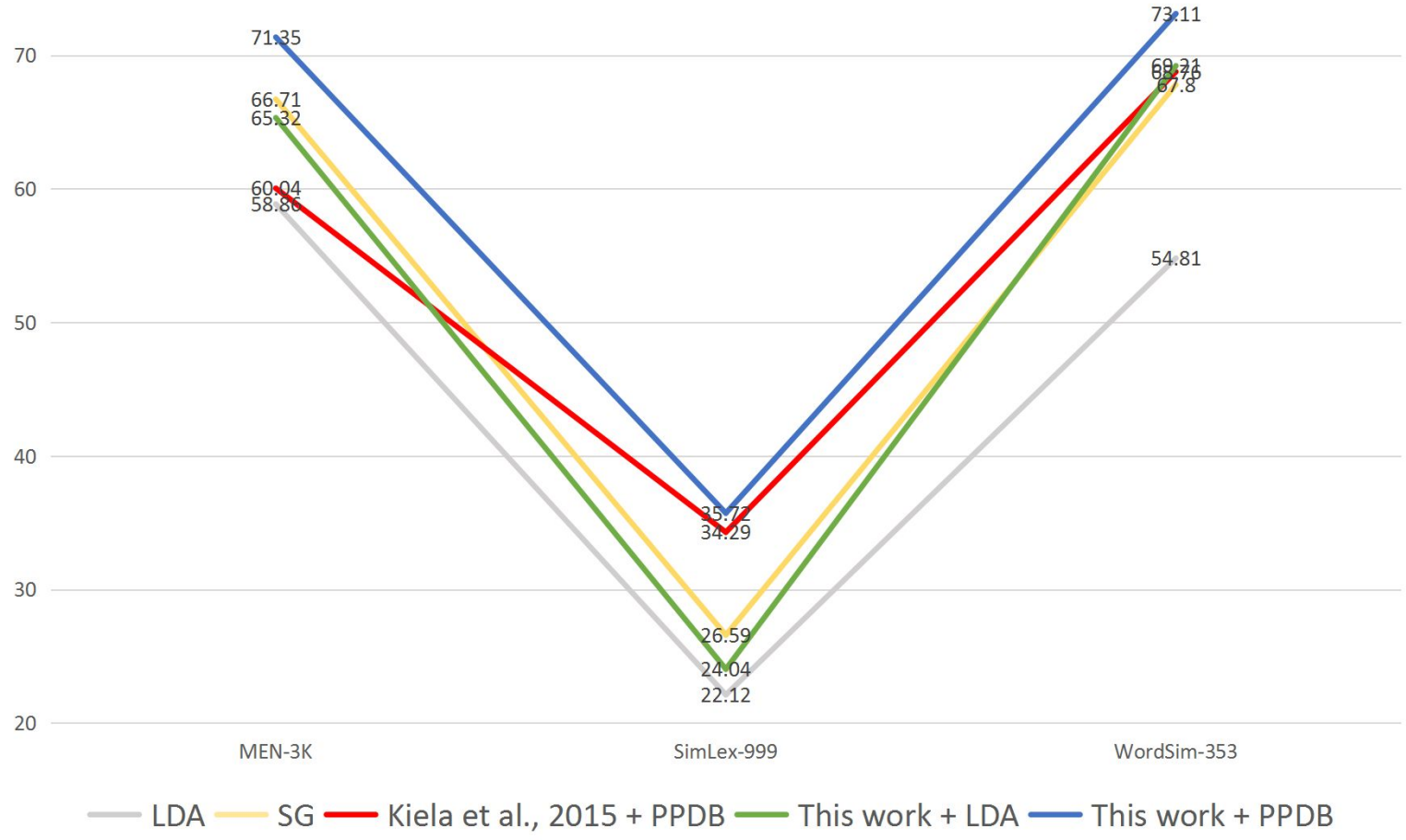
- A better way to learn and enhance word embeddings with:
  - Context
  - External knowledge
- A general framework with a regularizer, which can take into account:
  - Annotated knowledge, such as lexicon, dictionary
  - Unannotated knowledge, such as automatic clustered words

Regularization



- The regularization is performed on a matrix that represents external knowledge where:
- Each word is represented by a vector (row) in this matrix
- Each vector is normalized to the same length

Experimental Results - Word Similarity

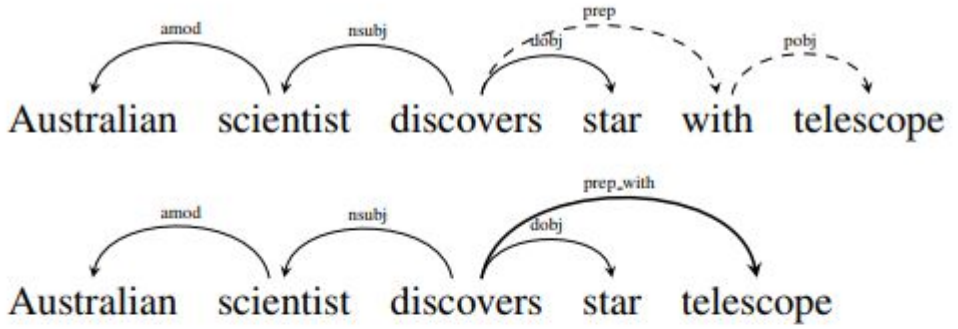


Extrenal Semantic Guidance

- Integrating context learning with human knowledge
- There are several ways to add such knowledge
  - After the embeddings are learned (retrofitting)
  - Joint learning
    - From X side
    - From Y side
- The quality of the knowledge is important

Structural Knowledge

- Levy and Goldberg (2014)
- Learning from dependency edges
- Automatically incorporate syntax and predicate-argument relations

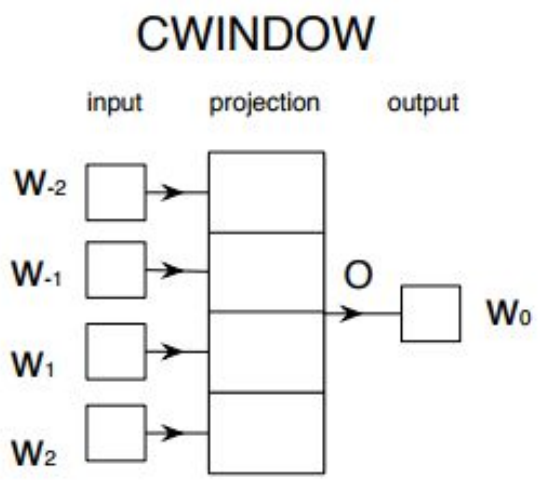


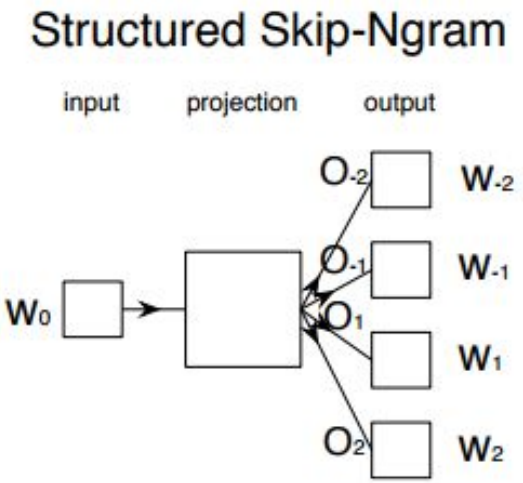
WORD	CONTEXTS
australian	scientist/amod^{-1}
scientist	australian/amod, discovers/nsubj^{-1}
discovers	scientist/nsubj, star/dobj, telescope/prep_with
star	discovers/dobj^{-1}
telescope	discovers/prep_with^{-1}

Figure 1: Dependency-based context extraction example. **Top:** preposition relations are collapsed into single arcs, making *telescope* a direct modifier of *discovers*. **Bottom:** the contexts extracted for each word in the sentence.

Word-order Knowledge

- Ling et al. (2015a)
  - Take structural information (word order) into account
  - Enlarging projection layer
  - Helps syntactic problem





Word-contribution Knowledge

- Ling et al. (2015b)
  - An attention enhanced CBOW
  - Weighted projection layer
  - Has a biased view of word v.s. context

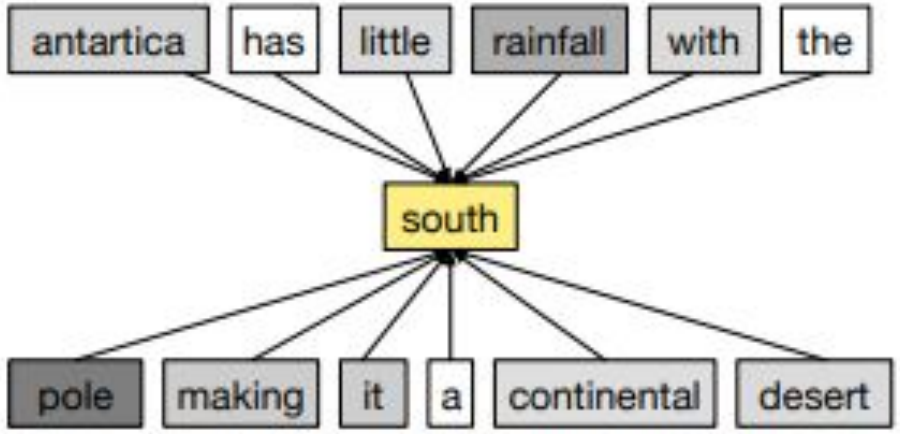


Figure 1: Illustration of the inferred attention parameters for a sentence from our training data when predicting the word *south*; darker cells indicate higher weights.

Word-order Knowledge

- Song et al. (2018)
  - CWindow, Structured SG, Attention CBOW are too complicated
  - The only information we need is the word order
  - Learning word order along with the context
  - Use an algorithm similar to negative sampling

Model	Parameters	Operations
SG	$2\|V\|d$	$2c\mathcal{C}(n+1)o$
SSG	$(2c+1)\|V\|d$	$4c^2\mathcal{C}(n+1)o$
SSSG	$3\|V\|d$	$4c\mathcal{C}(n+1)o$
DSG	$3\|V\|d$	$2c\mathcal{C}(n+2)o$

The Use of Language Structure

- To model exactly how words are associated
- Relative positions of words are important
- Learning from syntactic information is a solution, but expensive
- Word order is probably the only guidance one can obtain from running text other than introducing external knowledge
- Weak but effective, esp. for syntactic tasks

Sub-word components

- Luong (2013)
  - Decompose word with affix and stem
  - Two-layer structure

$$\boldsymbol{p} = f(\boldsymbol{W_m}[\boldsymbol{x}_{\texttt{stem}}; \boldsymbol{x}_{\texttt{affix}}] + \boldsymbol{b_m})$$

$$s\left(n_i\right) = \boldsymbol{v}^\top f(\boldsymbol{W}[\boldsymbol{x}_1; \ldots; \boldsymbol{x}_n] + \boldsymbol{b})$$

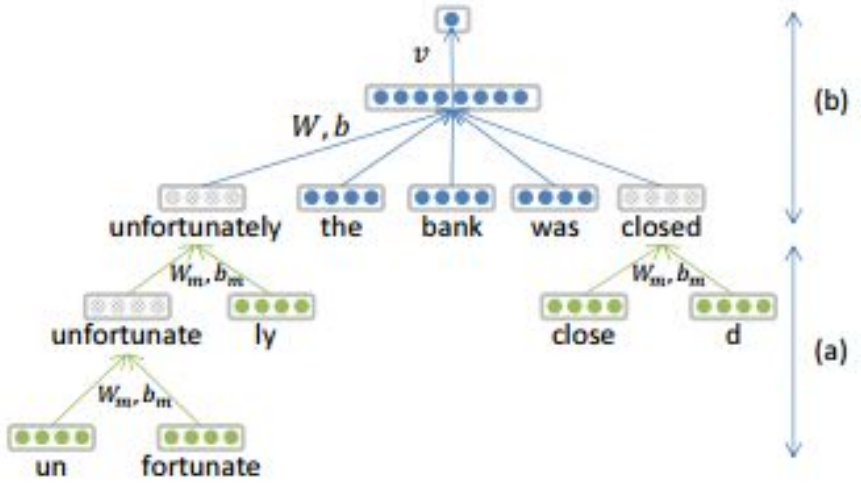
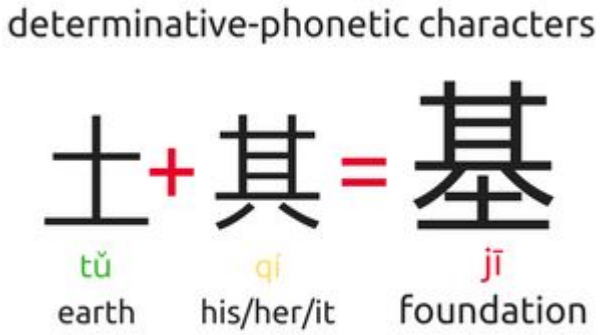
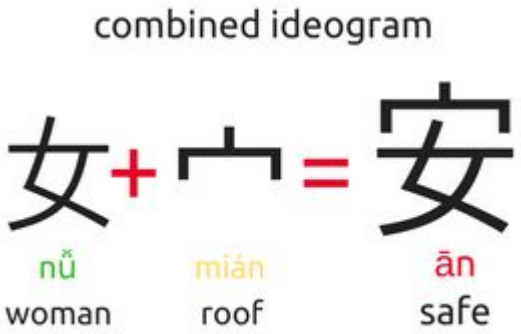


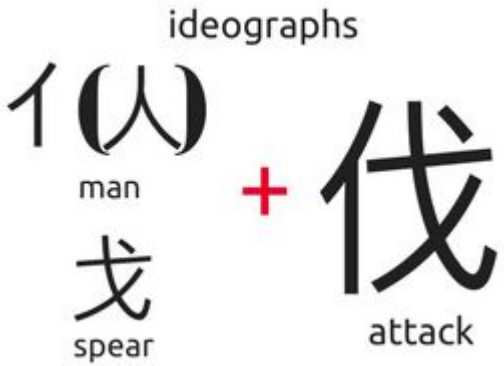
Figure 2: **Context-sensitive morphological RNN** has two layers: (a) the *morphological* RNN, which constructs representations for words from their morphemes and (b) the *word-based* neural language which optimizes scores for relevant ngrams.

Sub-word from Chinese

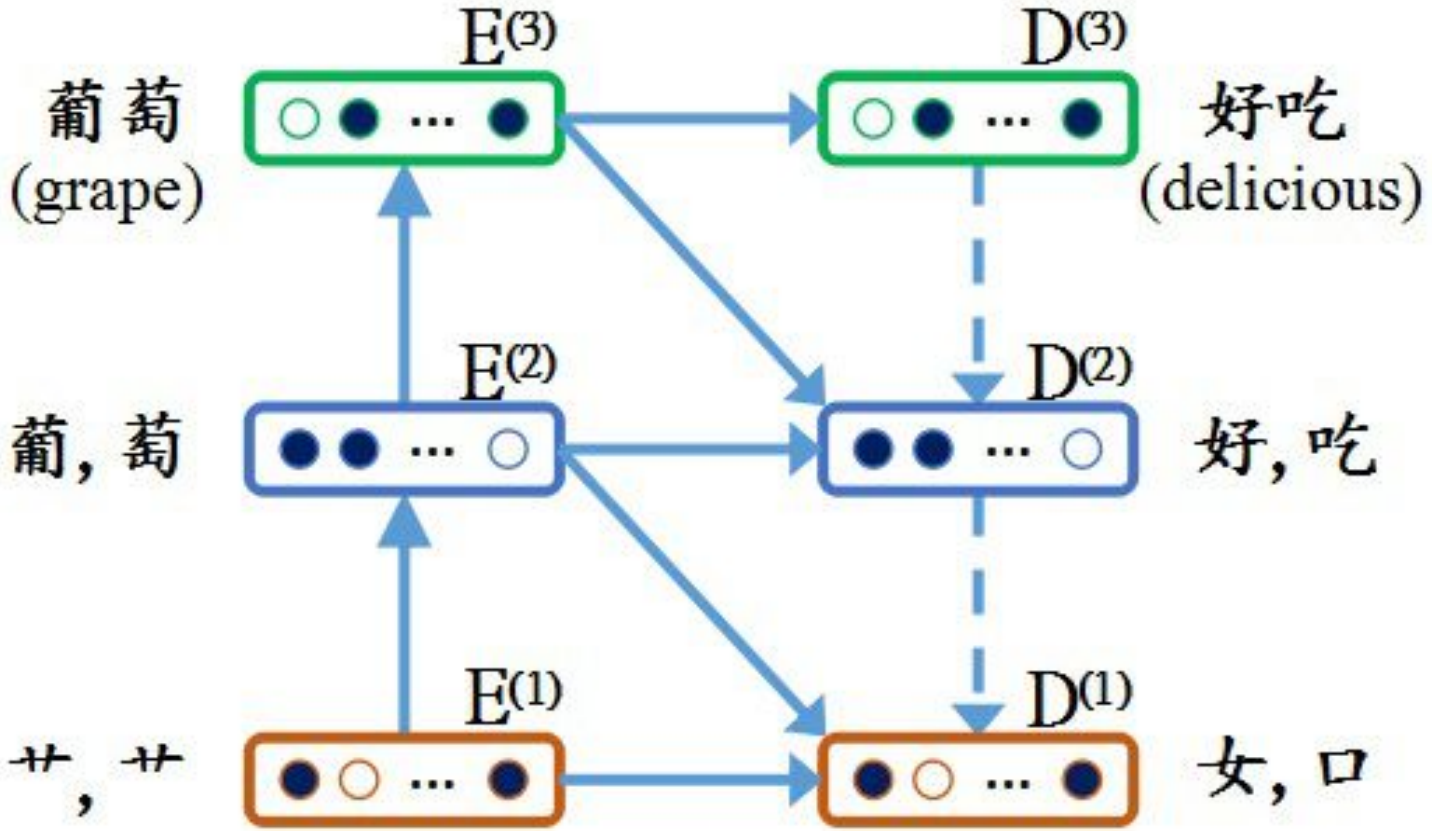
- Chinese characters can be decomposed into smaller semantic units, so do words
- This decompostion is an informative process







The Structure



- Components of Chinese words are hierarchically connected through a ladder to form a semantic derivation process.

Formulation

Optimization is to maximize the likelihood of all words over the corpus:

$$\mathcal{L}_V = \frac{1}{|V|} \sum_{w \in V} \mathcal{L}_{LSN}$$

where

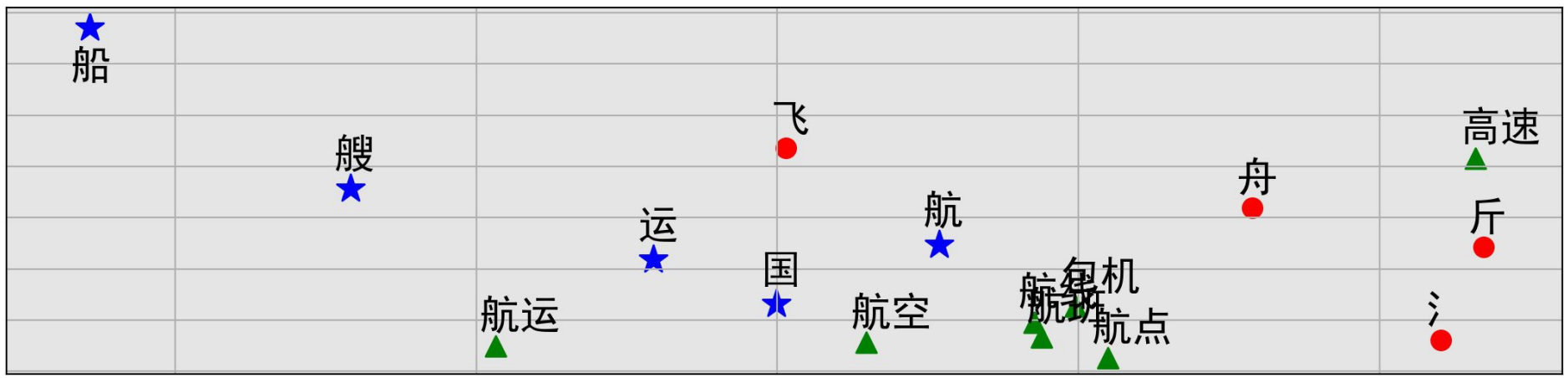
$$\begin{aligned} \mathcal{L}_{LSN} = & \mathcal{L}_{E^{(1)}E^{(2)}} + \mathcal{L}_{E^{(2)}E^{(3)}} + \mathcal{L}_{E^{(1)}D^{(1)}} \\ & + \mathcal{L}_{E^{(2)}D^{(2)}} + \mathcal{L}_{E^{(3)}D^{(3)}} \\ & + \mathcal{L}_{E^{(3)}D^{(2)}} + \mathcal{L}_{E^{(2)}D^{(1)}} \end{aligned}$$

Paritial derivatives are used to update each layer by SGD.

The overall effect of learning the entire model is to restrict all components of words in a unified vector space, different components can be clustered.

Results

Example	Nearest Words	Nearest Characters	Nearest Radicals
word: 航线 (airline)	航班 (flight), 包机 (charter flight), 航点 (waypoint)	航 (navigate), 国 (country), 运 (transport)	舟 (boat), 氵 (water), 斤 (half-kilogram),
character: 航 (navigate)	航线 (airline), 航空 (aviation), 航运 (shipping)	舱 (cabin), 艘 (ship), 船 (boat)	舟 (boat), 飞 (fly), 氵 (water)
radical: 舟 (boat)	高速 (high speed), 官兵 (officers and men), 战争 (war)	艘 (ship), 舰 (ship), 舷 (boat)	走 (to walk), 尤 (particularly), 斤 (half-kilogram)



Results

	WS-240	WS-296
CBOW	19.22	10.94
SG	19.58	11.73
CWE	19.29	10.71
SCWE	18.88	11.03
SCWE+M	20.23	10.82
MGE	18.55	9.75
LSN (W+R)	17.79	10.30
LSN (W+C)	29.38	13.90
LSN (W+C+R)	**34.23**	**18.23**

When trained on a small corpus 1/1000 of the Wikipedia

	WS-240	WS-296
CBOW	51.25	53.82
SG	51.91	54.05
CWE	51.75	53.64
SCWE	52.11	54.20
SCWE+M	52.85	55.26
MGE	53.13	53.33
LSN (W+R)	52.01	53.44
LSN (W+C)	53.47	55.58
LSN (W+C+R)	**54.14**	**57.04**

When trained on the entire Wikipedia

Sub Word Helps

- Looking in the opposite direction
  - Leveraging intrinsic knowledge in words
- Especially for some languages other than English
  - When the components are semantically meaningful
- Effective when training data is small
  - A good solution for cold-start scenario
- Useful way for mophological analysis
  - A potential method helps linguists

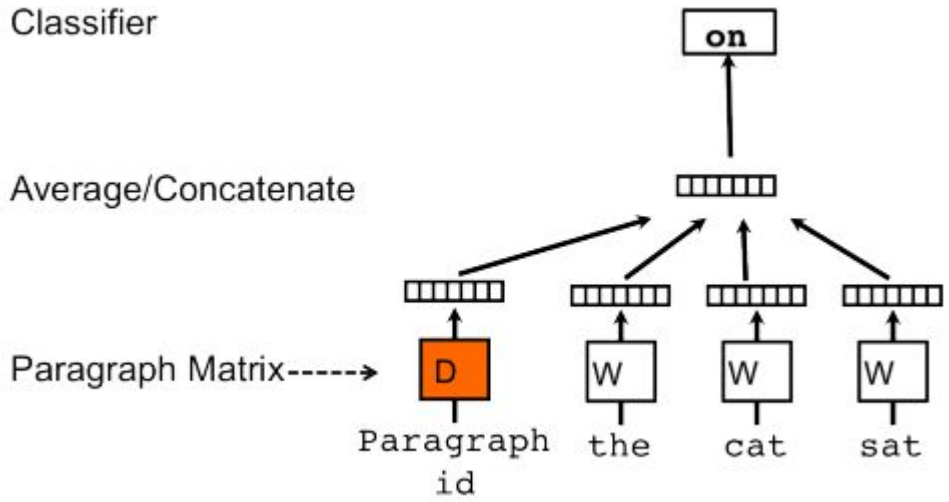
Extend to Phrase

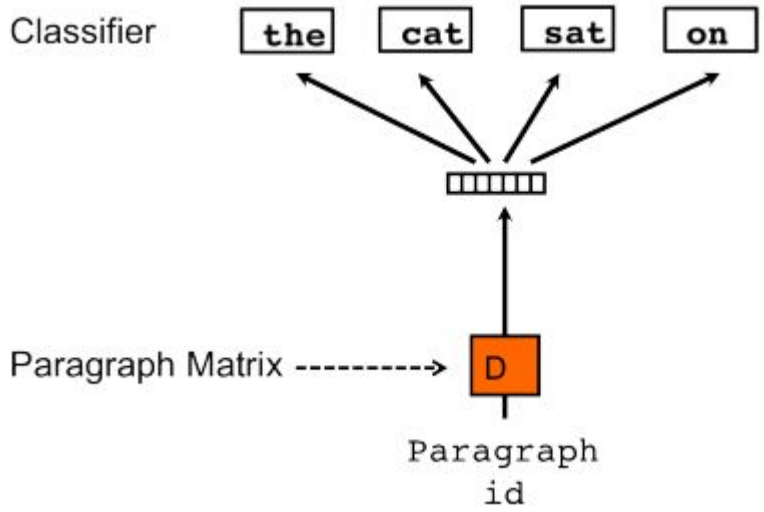
- Hill et al. (2016)
  - Sequence learning for phrase modeling
  - Use word embeddings as guidance to learn from dictionary
  - Pairwise-learning
    - Cosine similarity
    - Rank loss

$$\max(0, m - cos(M(s_c), v_c) - \cos(M(s_c), v_r))$$

Paragraph Vector

- Le and Mikolov (2014)
  - PV-DM
  - PV-DBOW
- Introduce a special token to represent paragraph
- Two structures correspond to CBOW and SG





Learning with a Target Task

- Maas et al. (2011)
- Learning word embeddings with sentiment labels
- A joint learning framework

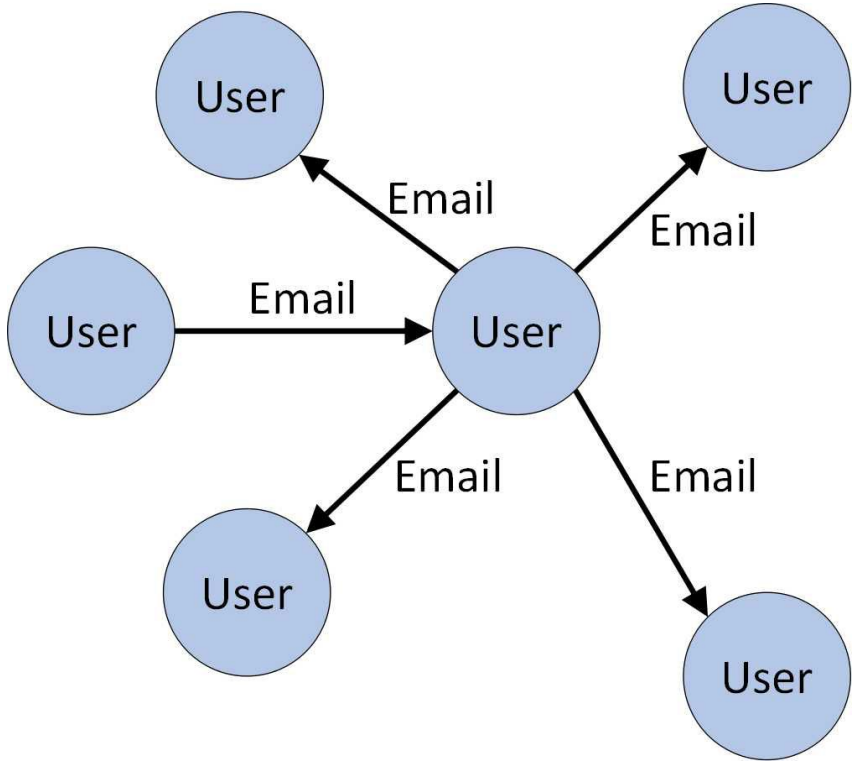
$$\nu||R||_F^2 + \sum_{k=1}^{|D|} \lambda||\hat{\theta}_k||_2^2 + \sum_{i=1}^{N_k} \log p(w_i|\hat{\theta}_k; R, b) + \sum_{k=1}^{|D|} \frac{1}{|S_k|} \sum_{i=1}^{N_k} \log p(s_k|w_i; R, \psi, b_c). \quad (11)$$

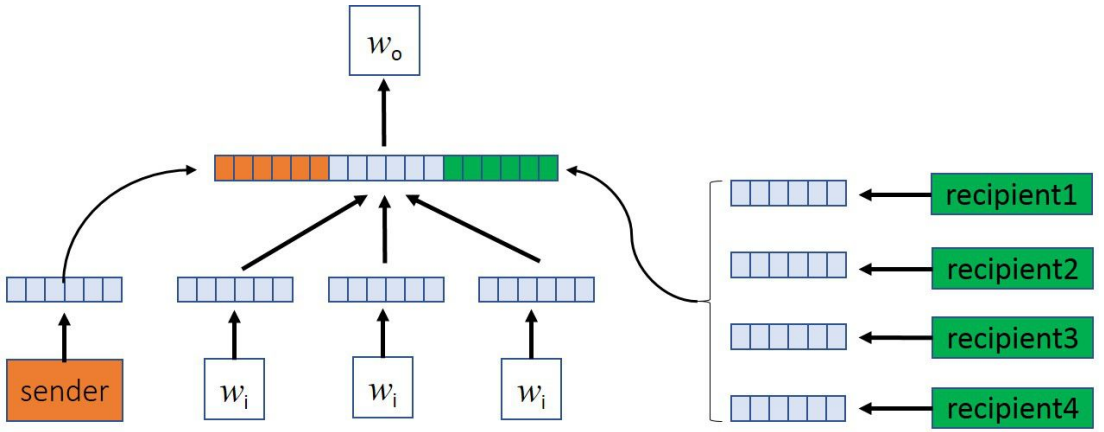
- A very early work

- You can try to learn your embeddings from its data

User Embedding

- Song and Lee (2017)





$$y = Xh(w_i, ..., w_{i+n};\ W, s, r_1, ..., r_m; U) + b$$

$$h = v_s \oplus \sum_{j=i}^{i+n} v_j \oplus \frac{1}{m} \sum_{r=1}^{m} v_r$$

Extensions

- Many possible options to apply the word2vec alike models
  - Many data or tasks have the nature similar to language modeling
- Embeddings is not only a result, but also a label for learning other language units
- Phrase/sentence is important for NLU
- Learning with (for) a task is the trend in the past few years

Hw5

- Prepare a presentation about sentence embeddings
  - Three groups
  - 40 mins per group
- Topics can be chosen from three papers
  - Kiros2015NIPS
  - Conneau2017EMNLP
  - Subramanian2018ICLR
- Slides should be done (email to me) by Monday noon (May 6th, 11:59am)

Hw5

- Contents of each presentation should include
  - The motivation of the study (why the model is designed in this way)
  - The algorithm (how the model is designed)
  - Implementation details (what makes the model work)
  - Performance (on what task and dataset)
  - Discussion (analysis)
- If possible, you can try the algorithm/data by youselves and include the results in your presentation