AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes

Speaker: Wenhu Chen

Author: Sascha Rothe and Hinrich Schutze

Background

- Word embedding is crucial for representation learning and downstream tasks like machine translation, sentiment analysis, etc.
- Glove, word2vec, and other word embedding can't learn embedding for synset and lexemes.
- This paper is motivated to learn embedding for synset and lexemes with pre-trained word embedding as inputs without relying on other corpus.

Background

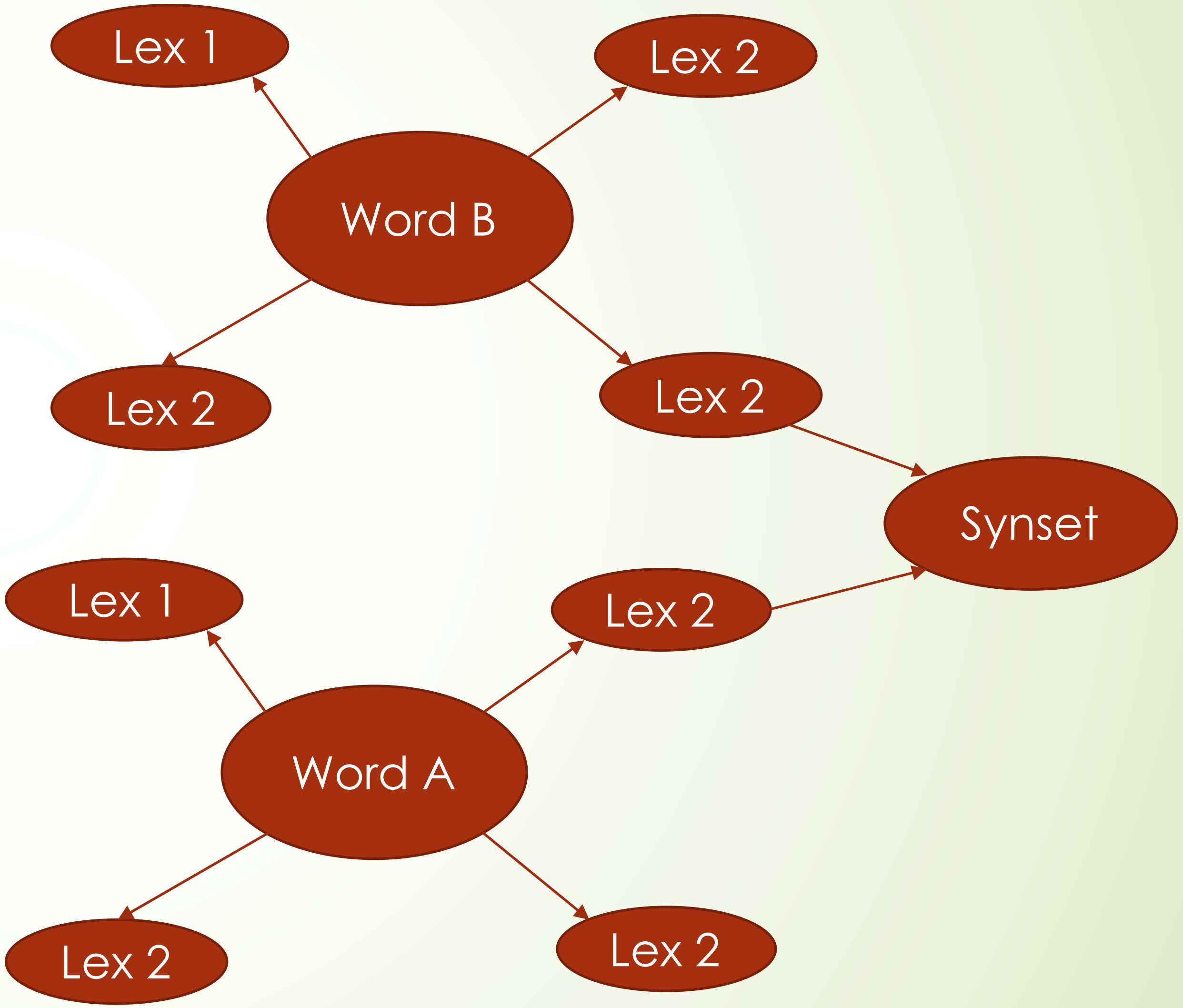
- Synset: A synset is a set of synonyms that are interchangeable in some context, e.g. (suit, case, lawsuit) belong to the synset of “lawsuit”.

- Lexeme: A particular spelling and its particular meaning, e.g. A word can have more than one lexeme, e.g. suit(cloth) and suit(law)have the same spelling form but are different lexemes.

Background

1. Words are composed of their lexemes.

2. Synsets are composed of their member lexemes.



Model

- Assumption 1: word embeddings are sums of lexeme embeddings

$$w^{(i)} = \sum_{j}^{N} l^{(i,j)}$$

- Assumption 2 : synset embeddings are sums of lexeme embeddings

$$s^{(j)} = \sum_{i}^{N} l^{(i,j)}$$

Note: $l^{(i,j)}$ is that lexeme of word $w^{(i)}$ that is a member of synset $s^{(j)}$

Model

- Encoder Model with parameter E

$$l^{(i,j)} = E^{(i,j)} w^{(i)}$$

$$s^{(j)} = \sum_i l^{(i,j)} = \sum_i E^{(i,j)} w^{(i)}$$

$$S = E \otimes W$$

Note: $\mathrm{E} \in R^{|S|*n*|W|*n}$ is a 4-D tensor, $W \in R^{|W|*n}$ and $\mathrm{S} \in R^{|S|*n}$ are 2D tensor
Encoder maps the word embedding into synset embedding

Model

- Decoder Model with parameter D

$$\bar{l}^{(i,j)} = D^{(i,j)} s^{(i)}$$

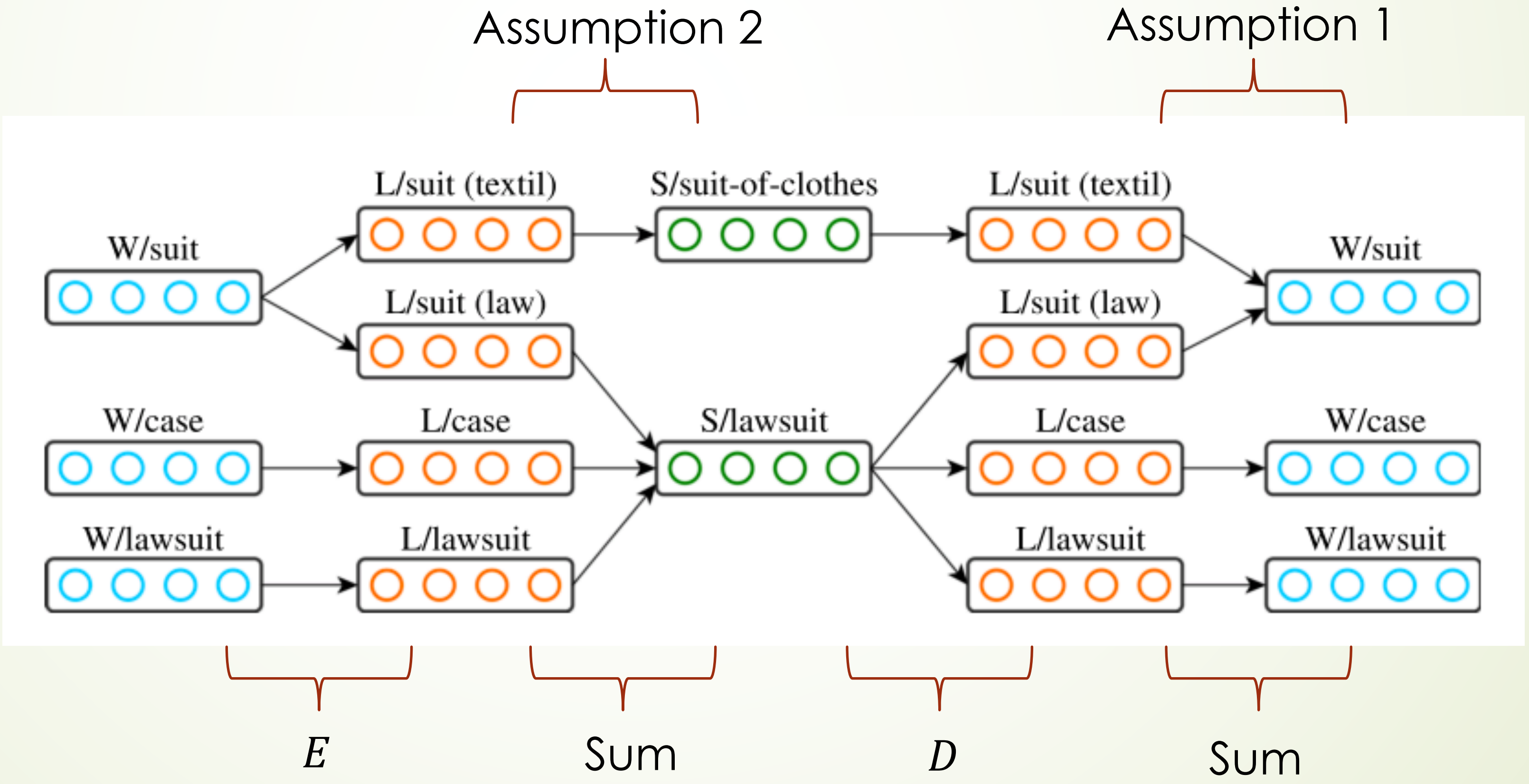
$$\bar{w}^{(i)} = \sum_j \bar{l}^{(i,j)} = \sum_j D^{(j,i)} s^{(j)}$$

$$\bar{W} = D \otimes S$$

Note: D $\in R^{|W|*n*|S|*n}$ is a 4-D tensor, $\bar{W} \in R^{|W|*n}$ and S $\in R^{|S|*n}$ are 2D tensor
Decoder maps the synset embedding back into word embedding

Model

- Autoencoder Architecture



Objective Function

- Synset Constraint: reconstructed word embedding should be consistent:

$$s^{(d)} = E^{(d)} w^{(d)} \qquad \bar{w}^{(d)} = D^{(d)} s^{(d)}$$

$$argmin_d \left|\left|D^{(d)} \otimes E^{(d)} \otimes w^{(d)} - w^{(d)}\right|\right| \quad \forall\, d$$

- Lexeme Constraint: lexemes should be consistent

$$l^{(i,j)} = E^{(i,j)} w^{(i)} \qquad \bar{l}^{(i,j)} = D^{(i,j)} s^{(i)}$$

$$argmin_{E,D} \left|\left|E^{(i,j)} w^{(i)} - D^{(j,i)} s^{(j)}\right|\right| \quad \forall i, j$$

Training

- Objective Function

$$Obj = \alpha \underbrace{||D^{(d)} \otimes E^{(d)} \otimes w^{(d)} - w^{(d)}||}_{\text{Synset Constraint}} + \beta \underbrace{||E^{(i,j)} w^{(i)} - D^{(j,i)} s^{(j)}||}_{\text{Lexeme Constraint}} + (1 - \alpha - \beta) Regularizer$$

- Dataset and Resource

$$\text{Word2vec} \xrightarrow{\text{autoExtend}} \text{Synset/Lex} \xrightarrow{\text{Evaluate}} \text{WSD/Similarity}$$

Experimental Results

			Senseval-2	Senseval-3
IMS feature sets	1	POS	53.6	58.0
	2	surrounding word	57.6	65.3
	3	local collocation	58.7	64.7
	4	S_{naive}-product	56.5	62.2
	5	S-cosine	55.5	60.5
	6	S-product	58.3	64.3
	7	S-raw	56.8	63.1
system comparison	8	MFS	47.6†	55.2†
	9	Rank 1 system	64.2†	72.9
	10	Rank 2 system	63.8†	72.6
	11	IMS	65.2‡	72.3‡
	12	IMS + S_{naive}-prod.	62.6†	69.4†
	13	IMS + S-cosine	65.1‡	72.4‡
	14	IMS + S-product	**66.5**	**73.6**
	15	IMS + S-raw	62.1†	66.8†
	16	IMS + $S_{optimized}$-prod.	*66.6*	*73.6*

WSD Accuracy Table

		AvgSim	AvgSimC
1	Huang et al. (2012)	62.8†	65.7†
2	Tian et al. (2014)	–	65.4†
3	Neelakantan et al. (2014)	67.2	69.3
4	Chen et al. (2014)	66.2†	68.9
5	words (word2vec)	66.6‡	66.6†
6	synsets	62.6†	63.7†
7	lexemes	**68.9**	**69.8**

Spearman Correction

This is the last slide
Thank you