CS898: Dynamic Word Embedding

By: XiaoLe (Eddie) Liu

Table of Content

1

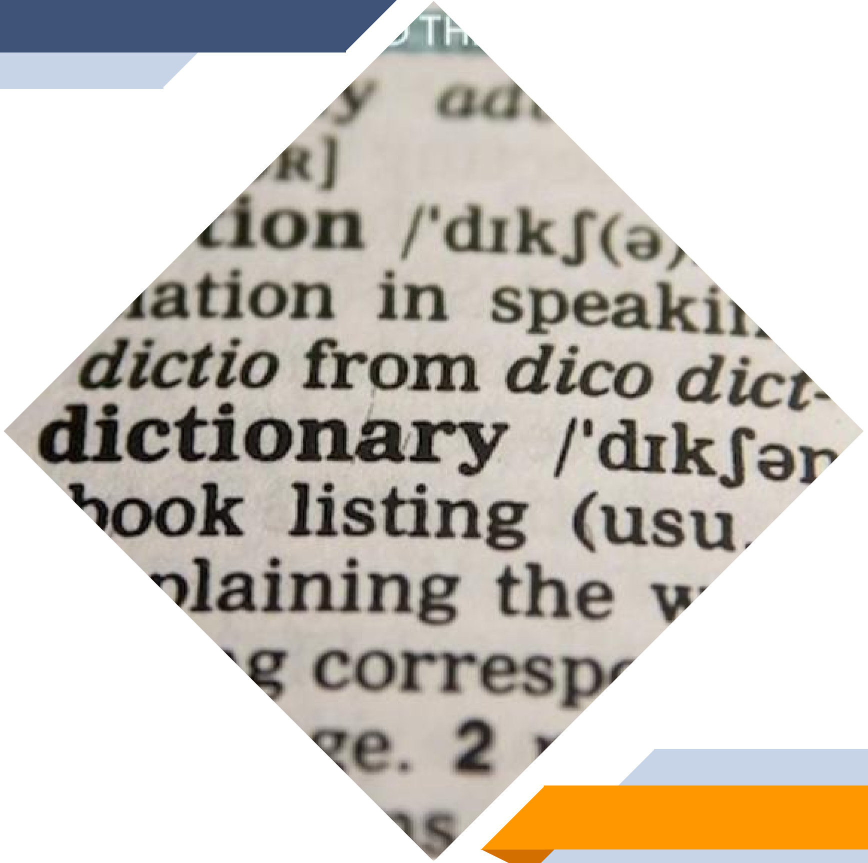
Word Embeddings & Dictionaries

Motivation: Word Vector Problems

- Rare words (e.g. min_count = n)
- Opposite words (e.g. good and bad)
- Words with multiple meaning (e.g. bat, ring, bush)

Intuition

During reading, when we do not understand a word, we look it up in the dictionary.



Understand Phrases by Embedding the Dictionary

- Idea: Train a Seq2Vec model using a word's dictionary definition as input and it's word vector as label
- Application: reverse dictionary and crossword puzzles
- Yoshua Bengio & TACL(2016)

Quantitative Result

Test Set		Dictionary definitions: Seen (500 WN defs)			Dictionary definitions: Unseen (500 WN defs)			Concept descriptions (200)		
Unsup. models	W2V add	-	-	-	923	.04/.16	163	339	.07/.30	150
	W2V mult	-	-	-	1000	.00/.00	10*	1000	.00/.00	27*
	OneLook	**0**	**.89/.91**	**67**	-	-	-	**18.5**	**.38**/.58	153
NLMs	RNN cosine	12	.48/.73	103	22	.41/.70	116	69	.28/.54	157
	RNN w2v cosine	19	.44/.70	111	19	.44/.69	126	26	**.38**/.66	111
	RNN ranking	18	.45/.67	128	24	.43/.69	103	25	.34/.66	102
	RNN w2v ranking	54	.32/.56	155	33	.36/.65	137	30	.33/.69	**77**
	BOW cosine	22	.44/.65	129	19	.43/.69	103	50	.34/.60	99
	BOW w2v cosine	15	.46/.71	124	**14**	**.46/ .71**	104	28	.36/.66	99
	BOW ranking	17	.45/.68	115	22	.42/.70	**95**	32	.35/.69	101
	BOW w2v rankng	55	.32/.56	155	36	.35/.66	138	38	.33/**.72**	85

median rank	*accuracy@10/100*	*rank variance*

Qualitative Result

Input Description	OneLook	W2V add	RNN	BOW
"a native of a cold country"	1:*country* 2:*citizen* 3:*foreign* 4:*naturalize* 5:*cisco*	1:*a* 2.*the* 3:*another* 4:*of* 5:*whole*	1:*eskimo* 2:*scandinavian* 3:*arctic* 4:*indian* 5:*siberian*	1:*frigid* 2:*cold* 3:*icy* 4:*russian* 5:*indian*
"a way of moving through the air"	1:*drag* 2:*whiz* 3:*aerodynamics* 4:*draught* 5:*coefficient of drag*	1:*the* 2:*through* 3:*a* 4:*moving* 5:*in*	1:*glide* 2:*scooting* 3:*glides* 4:*gliding* 5:*flight*	1:*flying* 2:*gliding* 3:*glide* 4:*fly* 5:*scooting*
"a habit that might annoy your spouse"	1:*sisterinlaw* 2:*fatherinlaw* 3:*motherinlaw* 4:*stepson* 5:*stepchild*	1:*annoy* 2:*your* 3:*might* 4:*that* 5:*either*	1:*bossiness* 2:*jealousy* 3:*annoyance* 4:*rudeness* 5:*boorishness*	1:*infidelity* 2:*bossiness* 3:*foible* 4:*unfaithfulness* 5:*adulterous*

Learning Word Embeddings from Intrinsic and Extrinsic Views

- Extrinsic: Context Information
- Intrinsic: Definitions & Explanations

Learning Word Embeddings from Intrinsic and Extrinsic Views

$$L = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0} \log\sigma(v_{w_t}^{T} v'_{w_{t+j}}) + \sum_{i=1}^{k}\mathbb{E}_{w_i} \sim P_n(w)[\log\sigma(-v_{w_t}^{T} v'_{w_i})],$$

VS

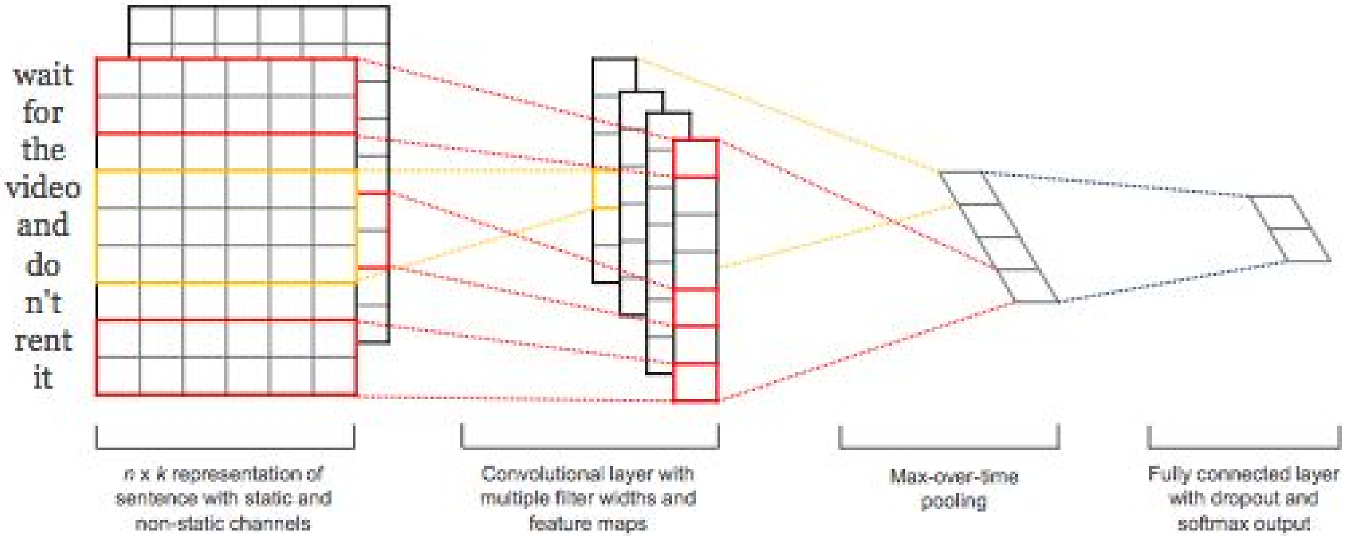
$$L = \sum_{-c\le j\le c, j\ne 0} \log\sigma(v_{w_t}^{T} v'_{w_{t+j}}) + \sum_{i=1}^{k}\mathbb{E}_{w_i} \sim P_n(w)[\log\sigma(-v_{w_t}^{T} v'_{w_i})] \boxed{+ \log\sigma(v_{w_t}^{T} v_{R(w_t)})}$$

Quantitative Result

	WS-353	MEN	MTurk-771	YP-130	SimLex-999
Skip-gram	44.57%	37.08%	31.95%	4.25%	17.88%
Glove	**45.35%**	32.93%	35.29%	8.64%	20.04%
DEWE	43.97%	**38.47%**	**36.93%**	**13.82%**	**21.46%**

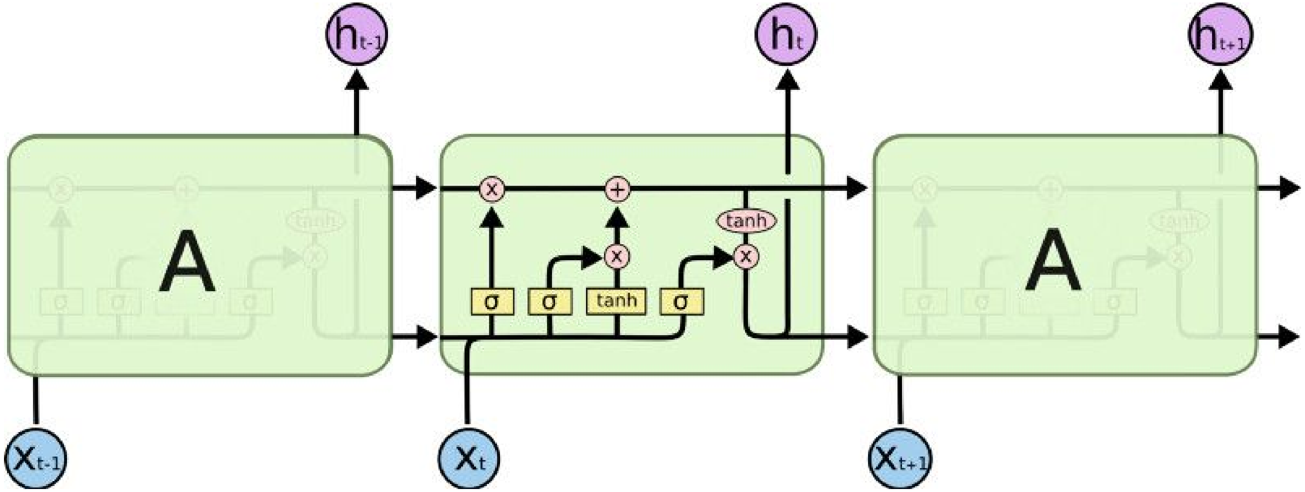
Def2Vec Models

- CNN (Kim Yoon,2014)
- LSTM (Original Paper)


wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

CNN

- Based on CNN from Yoon Kim (2014) paper on sentence classification
- Single convolutional layer with no pooling, and single fully connected layer.
- Zero padding to make definitions same length.
- Dropout after fully connected.



h_{t-1}
h_t
h_{t+1}
A
A
σ
σ
tanh
σ
tanh
X_{t-1}
X_t
X_{t+1}

LSTM

- Implemented with tf.contrib.rnn
- Final hidden state of LSTM as vector representation of the definition.
- Original paper have a linear mapping module at the end to convert internal state dimension to word vector dimension. Omitted here because, the internal state have the same dimension as word vector

Data

- Dictionary definition from multiple dictionaries (mainly WordNet)using Wordnik.com API
- Approximately **200,000** definitions of **75,000** words (original paper also trained on psuedo-definition from wikipedia, so it used Approximately 900,000 definition of 100,000 words)
- All word converted to **500**D word vectors using pre-trained Glove word embedding (Wikipedia + Gigaword 5). Unknown words randomly initialized.

Def2Vec Training Parameters

CNN

- **Input**: n*500 matrix
- **Window Size**: 3,4,5
- **Feature Map Size**: 500
- **Fully Connected Size**: 1024
- **Dropout Rate**: 0.5
- **Batch Size**: 50
- tf.train.**Adadelta**Optimizer

LSTM

- **Input**: sequence of 500D vectors
- **Internal State Size**: 500
- **Batch Size**: 16
- tf.train.**Adadelta**Optimizer

Loss Function

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

Evaluation

- 500 randomly picked Wordnet definitions
- Top 10 closest word vector according to cosine distance

37.1%

CNN accuracy @ top 10

39.8%

LSTM accuracy @ top 10

Interpretation of Result

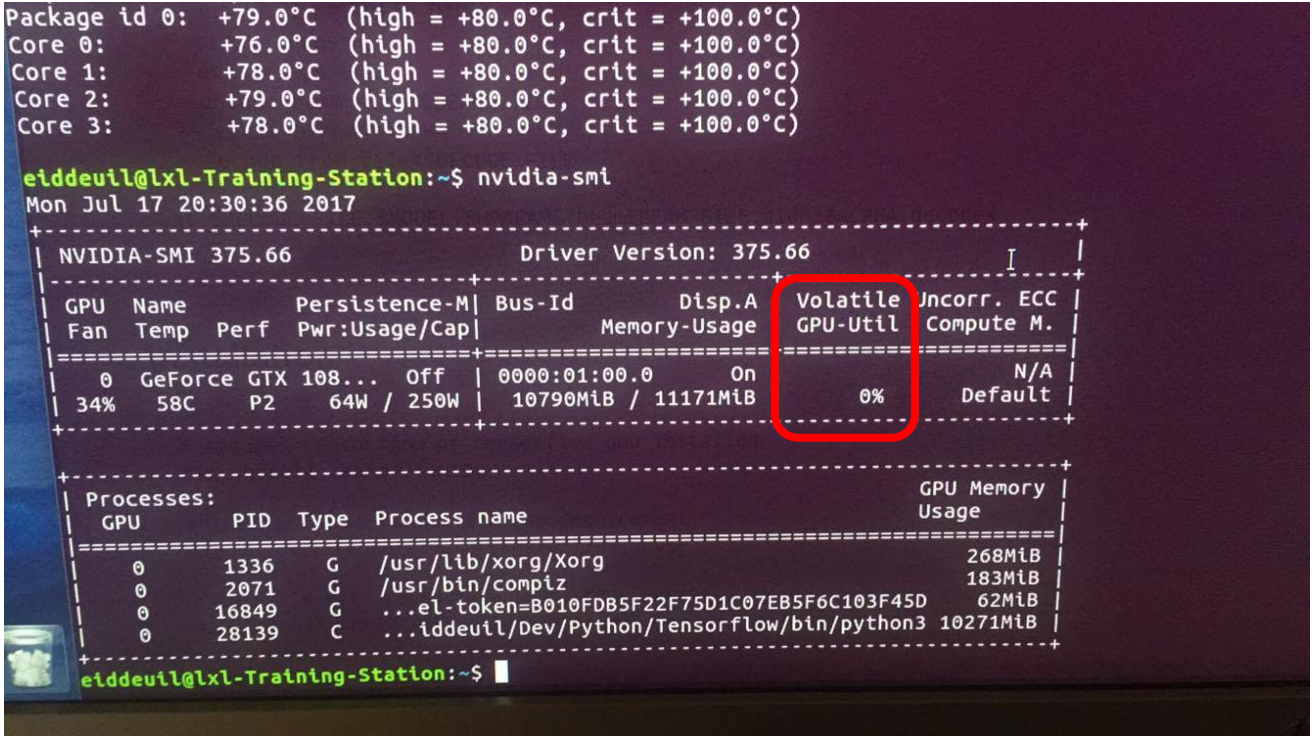
- Similar but worse than original result as expected, because of less training data.
- Overall, very poor accuracy, but prediction roughly in the region of correct word vector.
- Something is fundamentally wrong about this method.

Evaluation Methods for Unsupervised Word Embeddings

- Comprehensive study of evaluation methods
- Linguistic Insight:
  - Relatedness (similar = nearby?)
  - Coherence (group = related?)
- Downstream Tasks:
  - Sentiment Classification
  - Machine Translation

Tensor2Tensor

- Very new TensorFlow package that has a very modular architecture.
- State of the art models, datasets, and hyperparameters available
- Very efficient training on those models
- Problem: LSTM related model not working
- Trained Google’s Transformer model on NMT task.
  - 70% Accuracy, 90% Top5 on WMT_ENDE_8k data


Package id 0: +79.0°C (high = +80.0°C, crit = +100.0°C)
Core 0: +76.0°C (high = +80.0°C, crit = +100.0°C)
Core 1: +78.0°C (high = +80.0°C, crit = +100.0°C)
Core 2: +79.0°C (high = +80.0°C, crit = +100.0°C)
Core 3: +78.0°C (high = +80.0°C, crit = +100.0°C)
eiddeuil@lxl-Training-Station:~$ nvidia-smi
Mon Jul 17 20:30:36 2017
NVIDIA-SMI 375.66 Driver Version: 375.66
GPU Name Persistence-M Bus-Id Disp.A Volatile Uncorr. ECC
Fan Temp Perf Pwr:Usage/Cap Memory-Usage GPU-Util Compute M.
0 GeForce GTX 108... Off 0000:01:00.0 On N/A
34% 58C P2 64W / 250W 10790MiB / 11171MiB 0% Default
Processes: GPU Memory
GPU PID Type Process name Usage
0 1336 G /usr/lib/xorg/Xorg 268MiB
0 2071 G /usr/bin/compiz 183MiB
0 16849 G ...el-token=B010FDB5F22F75D1C07EB5F6C103F45D 62MiB
0 28139 C ...iddeuil/Dev/Python/Tensorflow/bin/python3 10271MiB
eiddeuil@lxl-Training-Station:~$

2

Improvement Idea

Any ideas?

Closer Look at Word Embedding Algorithms

- Context based
- Implicitly or explicitly use co-occurrence matrix
- One vector per word

Analysis & Problem

- Based on our data, one word have more than 2 definitions on average
- Context-based methods do not distinguish different meaning
- Definition must be precise, but the word vector might not be
- **Problem: Words with multiple word senses**

Resources

- WordNet: large lexical database of English
  - Grouped words that links to other groups
  - Human generate word sense: short definitions
- S.Arora Paper: Linear Algebra Structure of Word Senses, with Applications to Polysemy

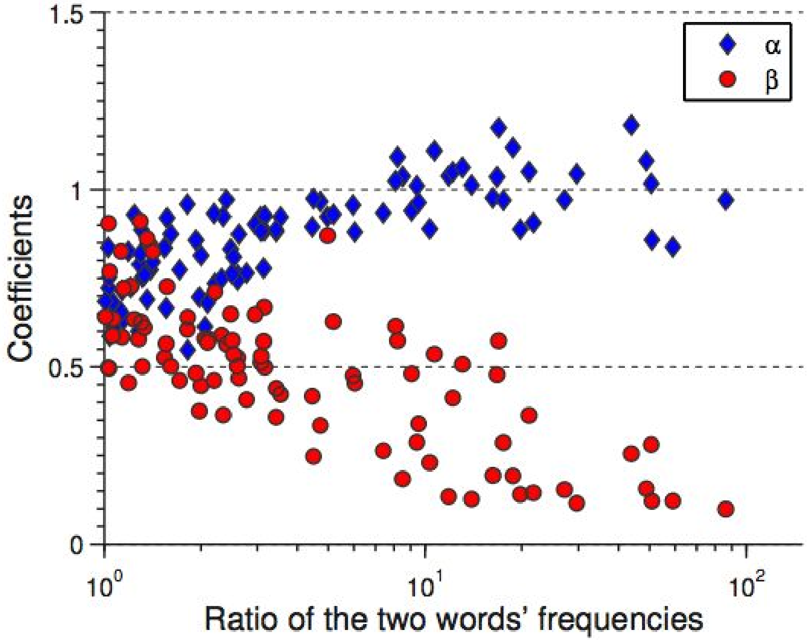
Word Senses & Atoms of Discourse

- Directions in word embedding represent topic or discourse
- Compare unit vector of topic to get their similarity
  - Unit vector because we use dot product to compute similarity
  - Look similar to human if dot product > 0.85, different if < 0.5
- The whole embedding is comprised of m topics

Thought Experiment

$$v_{w_{new}} \approx \alpha v_{w_1} + \beta v_{w_2}$$

$$\beta \approx 1 - c \lg r$$



Extracting Word Senses: Problem Formulation

Given word vectors in $\Re^d$, *totaling about* 60,000 *in this case, a sparsity parameter* k, *and an upper bound* m, *find a set of unit vectors* $A_1, A_2, \ldots, A_m$ *such that*

$$v_w = \sum_{j=1}^{m} \alpha_{w,j} A_j + \eta_w \tag{3}$$

where at most k *of the coefficients* $\alpha_{w,1}, \ldots, \alpha_{w,m}$ *are nonzero (so-called* hard sparsity constraint*), and* η_w *is a noise vector.*

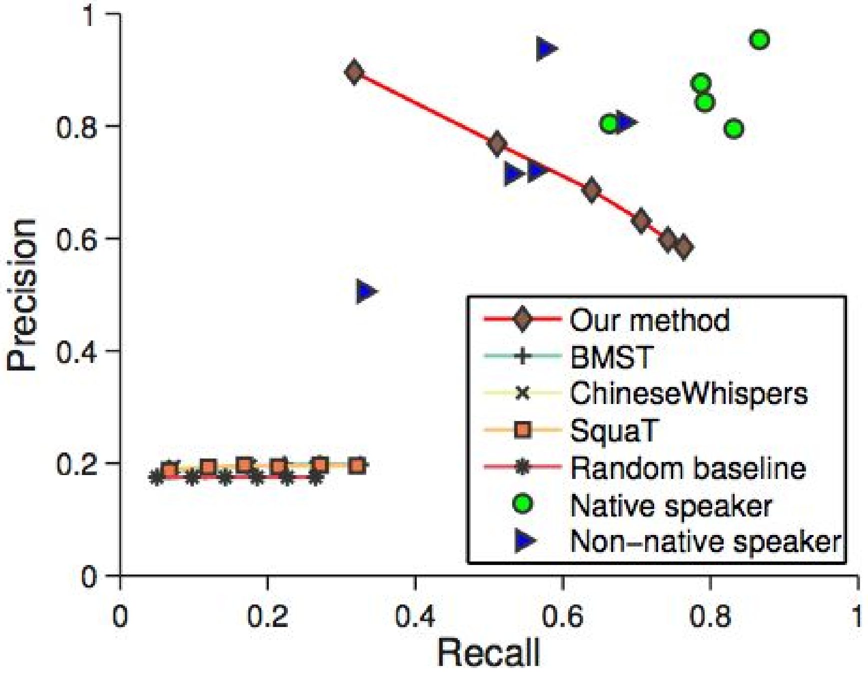
Extracting Word Senses: K-SVD

$$\sum_{w} |v_w - \sum_{j=1}^{m} \alpha_{w,j} A_j|_2^2.$$

Or

$$\min_{D,X}\{\|Y - DX\|_F^2\} \qquad \text{subject to} \quad \forall i, \|x_i\|_0 = 1.$$

Results



spring				
beginning	dampers	flower	creek	humid
until	brakes	flowers	brook	winters
months	suspension	flowering	river	summers
earlier	absorbers	fragrant	fork	ppen
ycar	whccls	lilics	pincy	warm
last	damper	flowered	elk	temperatures

tie				
trousers	season	scoreline	wires	operatic
blouse	teams	goalless	cables	soprano
waistcoat	winning	equaliser	wiring	mezzo
skirt	league	clinching	electrical	contralto
slccvcd	finishcd	scorclcss	wirc	baritonc
pants	championship	replay	cable	coloratura

Results

- 乒乓球，羽毛球，跳水，游泳，举重
- Ping pong, Badminton, Diving, Swimming, Lifting
- 藏羚羊，大熊猫，娃娃鱼，金丝猴，小熊猫
- Tibetan Antelope, Panda, Chinese Giant Salamander, golden monkey, small panda
- 人，球，大，小，一
- Person, Ball, Big, Small, One

Improvement on Previous Model

Use K-SVD to convert base embedding into atoms of discourses, and train with discourse vector as label.

Challenge in Experiment

- Takes up a huge amount of memory
- Segmentation Fault on big embeddings
- Non-Convex problem
- A lot of manual work required for preparing training data that match definition with word sense in terms of discourse vector.

3

Discussion

Dynamic Embedding Tool

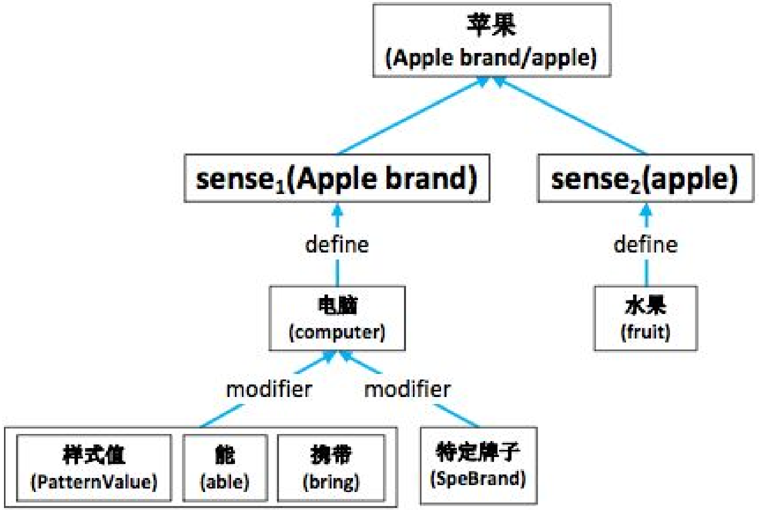
- Word Embedding Tools:
  - Add
  - Swap
  - Filter/Delete

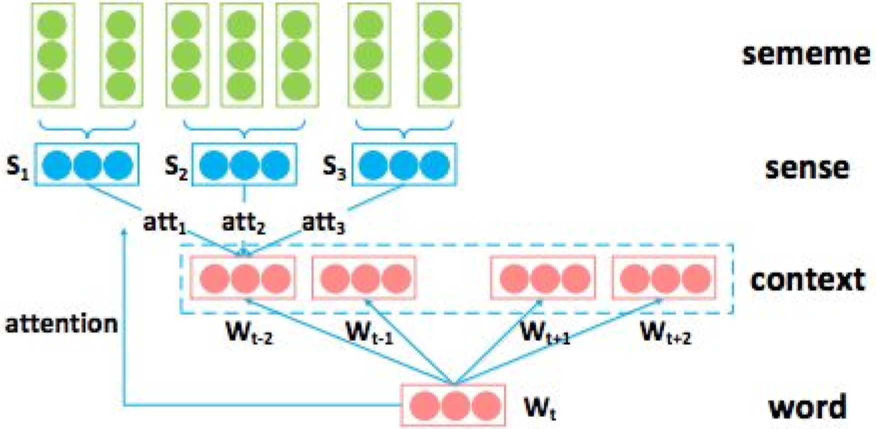
Possible Application: Chat

- Add: ask for clarification on novel concept/words and encode reply into word embedding to maintain conversational context.
- Swap: Convert polysemy words into multiple vectors to improve semantic accuracy
- Filter: Control grammar, rhyme, mood.

Related Work: Sememe (ACL 2017)

- Sememe: unit meaning
- HowNet (Chinese)
- Compose unit meaning to improve word embedding
- Model: attention over context words or target


苹果
(Apple brand/apple)
sense1(Apple brand)
sense2(apple)
define
define
电脑
(computer)
水果
(fruit)
modifier
modifier
样式值
(PatternValue)
能
(able)
携带
(bring)
特定牌子
(SpeBrand)


sememe
S1
S2
S3
sense
att1
att2
att3
context
attention
Wt-2
Wt-1
Wt+1
Wt+2
Wt
word

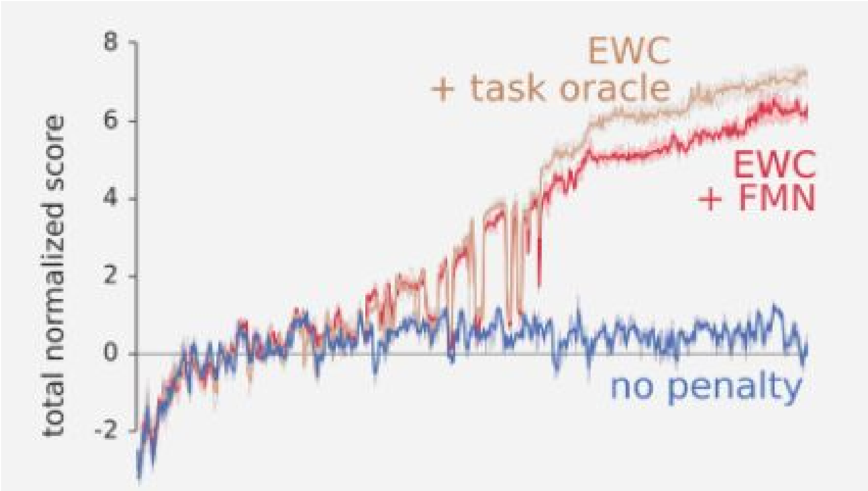
Related Work: Jump LSTM (ACL 2017)

- Mimic skimming
- Trained using reinforcement learning
- Improved accuracy of LSTM over long text.

Seq length	LSTM-Jump	LSTM	Speedup
Test accuracy			
10	**98%**	96%	n/a
100	**98%**	96%	n/a
1000	**90%**	80%	n/a
Test time (Avg tokens read)			
10	**13.5s (2.1)**	18.9s (10)	1.40x
100	**13.9s (2.2)**	120.4s (100)	8.66x
1000	**18.9s (3.0)**	1250s (1000)	**66.14x**

Related Work: Elastic Weight Consolidation

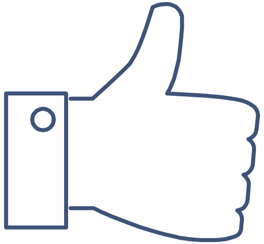
- Prevent forgetting, continual learning using experience from related tasks
- Intuitively similar to human brain's synaptic consolidation (proven neuroscience concept).



CREDITS

Special thanks to:

- Professor Li
- Junnan Chen
- Wordnik.com
- Presentation template by SlidesCarnival



THANKS!

Any questions?