IMPROVING QUALITY OF FEATURES IN VECTOR EMBEDDING MODELS BY USING SYNSETS

Yves Greatti Utku Evci

FML2016 - Foundations of Machine Learning Class

20^{th} December 2016

Introduction

- Neural language modeling (NLM) have been successful in capturing semantic information in semi-supervised settings
- Word2vec is a shallow NNet which generates word embedding vectors in low-dimensional space
- Word2vec has two architectures: CBOW or Skip-Gram
- Semantically close words correspond to vectors with similar algebraic properties such as: ”v[man] - v[king] + v[queen] $\approx$ v[woman]”

Motivation

- word ambiguity, an important part of a language, is ignored
- Words can have more than one meaning or ”sense”
- ”break” has 75 meanings, ”bank” 10 different meanings as a noun, and 8 as a verb
- Be able to have a language model discerning these meaning can benefit NLP tasks like: text classification, NER, sentiment analysis or text summarization

Wordnet

- Wordnet: a large lexical database of English words
- Nouns, verbs, adverbs and adjective are grouped into sets of distinct concepts called ”synsets”
- WordNet 3.0 statistics: 117,798 nouns, 11,529 verbs, 22,479 adjectives, and 4,481 adverbs. The average noun has 1.24 senses, and the average verb has 2.17 senses.
- Relations between synsets: hyperonymy/hyponymy, Meronymy/holonymy, antonymy

Wordnet

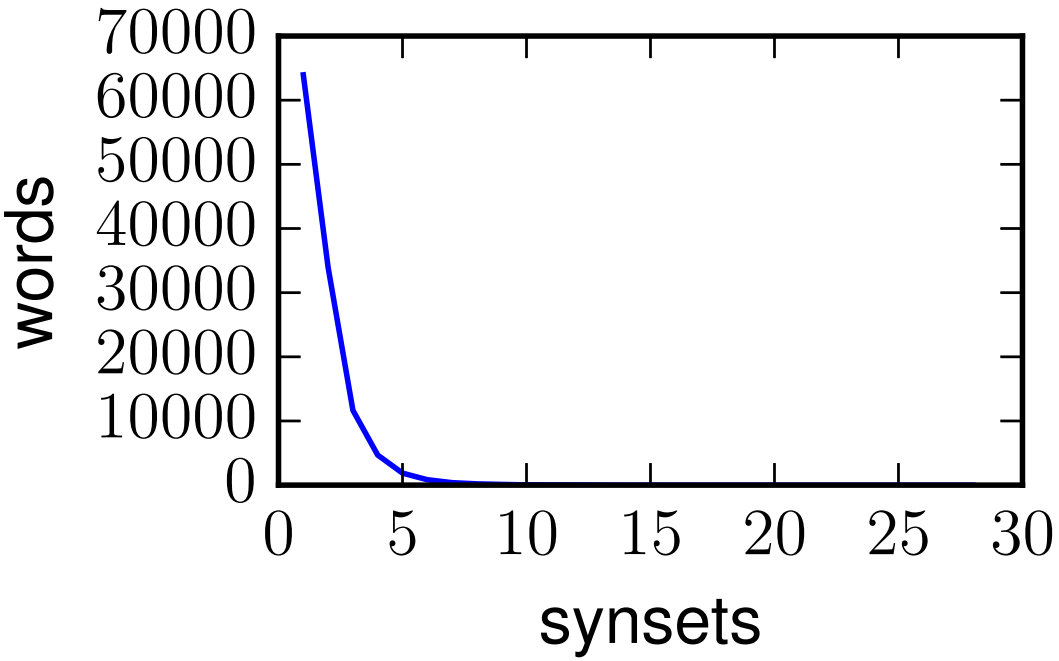

Figure 1: Synsets vs. words distributions

Experimentation Components

- python script wordnet_utils.py: lemmatizes and tags input corpus using from nltk: WordnetLemmatizer and Perceptron tagger
- WSD a wrapper of DKPro WSD a java word disembiguation framework, and more specifically it uses the Extended Lesk algorithm
- TensorFlow word2vec_optimized which is a multi-threaded w2v unbatched skip-gram (we have also a lighter version syn2vec)
- Performing tasks in Convolutional Neural Networks for Sentence Classification[1] with new word embeddings

[1] Y. Kim (2014)

Results

Google's data set: 19,544 questions

Epoch	15	150	300
word text8	36.5% (6511/17827)	39.3% (7005/17827)	39.6% (6888/17827)
synset text8	57.4% (9459/16479)	61.5% (10075/16479)	60.6% (9988/16479)

Table 1: Accuracy for word and synset models

	tokens	unique tokens	unique frequent tokens
words	17005207	253854 (-1.49%)	71290 (-28%)
synsets	17005200	306620 (-1.80%)	99004 (-32%)

Table 2: Vocabulary size

Results

	word text8	word text8	word text8	synset text8	synset text8	synset text8
Epoch	15	150	300	15	150	300
capital-common-countries	82.4 (417/506)	85.6 (433/506)	82.6 (418/506)	47.8 (242/506)	55.1 (279/506)	58.1 (294/506)
capital-world	39.2 (1396/3564)	46.3 (1651/3564)	46.8 (1668/3564)	24.8 (767/3091)	31.5	32.4 (1000/3091)
currency	11.9 (71/596)	13.1 (78/596)	9.7 (58/596)	5.7 (26/460)	2.6 (12/460)	1.5 (7/460)
city	39.1 (912/2330)	48.1 (1120/2330)	46.7 (1088/2330)	18.9 (386/2046)	28.7 (588/2046)	26.1 (533/2046)
family	46.7 (196/420)	44.0 (185/420)	48.1 (202/420)	28.3 (107/378)	35.2 (133/378)	33.3 (126/378)
gram1-adj-adv	7.7 (76/992)	7.2 (71/992)	6.1 (61/992)	2.8 (26/928)	4.1 (38/928)	4.1 (38/928)
gram2-opposite	6.6 (50/756)	6.2 (47/756)	7.9 (60/756)	3.7 (17/454)	3.1 (14/454)	1.1 (5/454)
gram3-comparative	48.6 (648/1332)	40.8 (543/1332)	41.3 (550/1332)	100 (1322/1322)	100 (1322/1322)	100 (1322/1322)
gram4-superlative	20 (198/992)	18.2 (181/992)	15.6 (155/992)	97.1 (1086/1118)	97.1 (1086/1118)	97.1 (1086/1118)
gram5-present-participle	21.5 (227/1056)	26.9 (284/1056)	22.9 (242/1056)	96.9 (1016/1048)	96.9 (1016/1048)	96.9 (1016/1048)
gram6-nationality-adj	74.2 (1129/1521)	77.1 (1172/1521)	76.4 (1162/1521)	61.9 (895/1445)	72.2 (1043/1445)	68.7 (992/1445)
gram7-past-tense	26.9 (419/1560)	24.9 (388/1560)	25.6 (400/1560)	95.1 (1472/1548)	95.1 (1472/1548)	95.1 (1472/1548)
gram8-plural	42.5 (566/1332)	48.3 (644/1332)	46.8 (624/1332)	97.0 (1231/1269)	97.0 (1231/1269)	97.0 (1231/1269)
gram9-plural-verbs	23.7 (206/870)	23.9 (208/870)	23.0 (200/870)	100 (866/866)	100 (866/866)	100 (866/866)

Table 3: Accuracy for word and synset models per question category

Results: Nearest Neighbors of *net*

word model	*cosine similarity*	*synset model*	*cosine similarity*
net	1.0000	internet%1:06:00::	1.0000
http	0.5305	internet	0.7942
org	0.5211	ip%1:09:00::	0.6143
com	0.5189	protocol%1:10:01::	0.5777
www	0.5164	ip	0.5731
migration	0.4536	provider%1:18:00::	0.5720
https	0.4498	user%1:18:02::	0.5694
html	0.4440	web%1:05:01::	0.5597
website	0.4411	isps	0.5583
online	0.4411	network%1:06:03::	0.5483
htm	0.4395	server%1:18:01::	0.5450
irc	0.4380	arpanet	0.5388
migrant	0.4250	rfc	0.5373
wiki	0.4099	network%1:06:02::	0.5342
database	0.4078	mail%1:10:00::	0.5285
forum	0.4053	wireless%3:00:00::	0.5256

Table 4: Nearest Neighbors of *net*

Sentence Classification

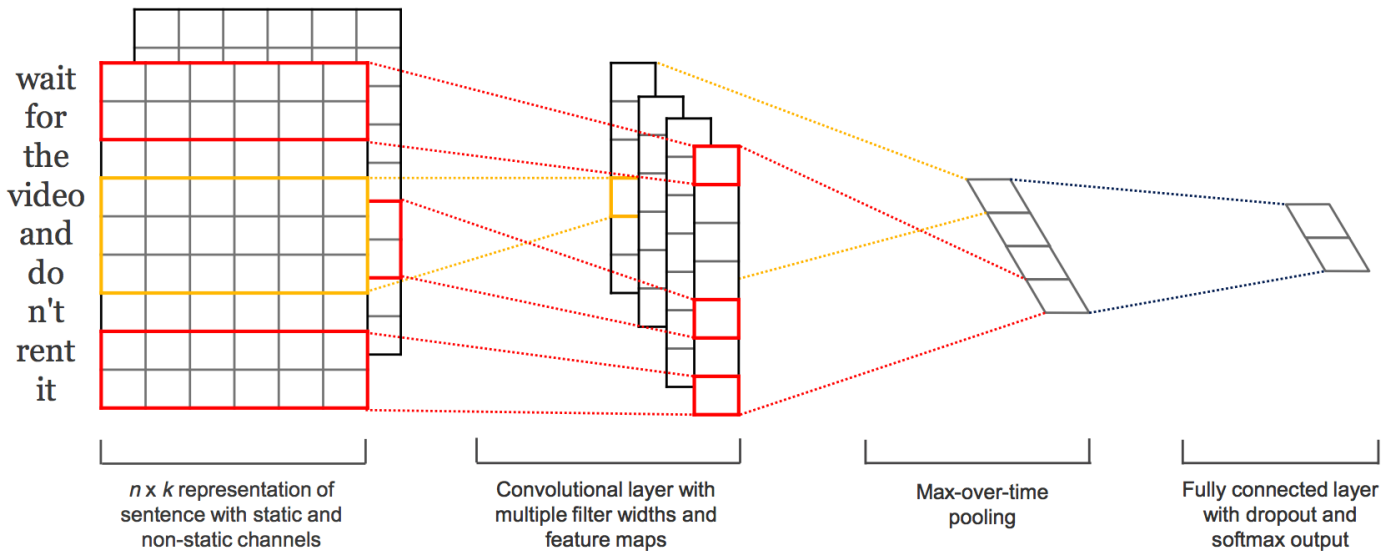

Figure 2: Convolutional model using word embeddings for sentence classification

Sentence Classification: Results

Task	random	word2vec(static)	syn2vec(static)	syn2vec(non-static)
MR	75.9	80.5	69.2	72.5
SST1	42.2	44.8	35.9	41.9
SST2	83.5	85.6	73.1	80.2
SUBJ	89.2	93.0	88.2	89.0
CR	78.3	83.3	75.0	77.4
MPQA	84.6	89.6	84.7	84.3
TREC	88.2	91.8	90.0	88.8

Table 5:
Accuracy of different models on sentence classification tasks

Conclusion

- Overall accuracy of Synset model good but per category not great
- Needs new test dataset in term of exact meanings
- Concept justified but too many components involved
- Next step: since word2vec, tagging and WSD involve a word context, combine them in a new model with loss function on the norm of the synset embedding vectors
- code: https://github.com/citrucel/FML-FA16-Project