Word Representation

13-ACL16-Yandex-Siamese CBOW: Optimizing Word Embeddings for Sentence Representations

PDF, Bib, Theono

Siamese CBOW trains word embedding directly for the purpose of being averaged to produce better sentence representation.

One question need to be answered: **How to select negative samples? random?** according to similarity? refer to the code!
Update: Both in word2vec[line 442-446] & Siamese CBOW[line 432-448], random slect negative samples!

- [**Model**]
  - Constructing a supervised training criterion by having our network predict sentences occuring next to each other in the training data, which is similar to **Skip-thought**.
- [**Related Work**]
  - Word2vec
  - Skip-thought
- [**Training Set**]
  - Toronto Book Corpus: 74,004,228 setences; 1,057,070,918 tokens, originating from 7087 unique books.
  - consider tokens appearing 5 times or more, which leads to a vocabulary of 315,643 words.
  - http://www.cs.toronto.edu/~mbweb
- [**TODO**]
  - **replace word2vec in STS with this model**

18-ACL16-PKU-Compressing Neural Language Models by Sparse Word Representations

PDF, Bib, keras

- [**Problem**]: a). Memory-consuming b).neural LMs are unlikely to learn meaningful representations
- [**Motivation**]: In a dictionary, an unfamiliar word is typically defined by common words.
- [**Model**]:
  - To learn the sparse codes, "true" embeddings by SkipGram for both common words and rare words. However, these true embeddings are slacked during our language modeling.
  - Parameter Compression for Embedding Subnet
  - Parameter Compression for Prediction Subnet -- share the same set of sparse codes to represent word vectoes in Embedding and the output weights in the Prediction Subnet(this is an assumption).
  - Noise-Contrastive Estimation with ZRegression(not useful)
- [**Experiments**]:
  - Dataset
  - Qualitative Analysis
  - Quantitative Analysis (Setting and Performance)
  - Effect of some component
- [**Remark**]:

  - This paper is a traditional writting skills, INTRODUCTION(Question, Motivation, Contribution). BACKGROUND. MODEL(several parts). EXPERIMENTS(This part is well written). CONCLUSION.
  - ^1 is another Sparse Word Vector Representations. It solve an optimization problem to obtain the sparse vector of words as well as a dictionary matrix simulataneously.

  ^1. 18-ACL15-CMU-Sparse Overcomplete Word Vector Representations

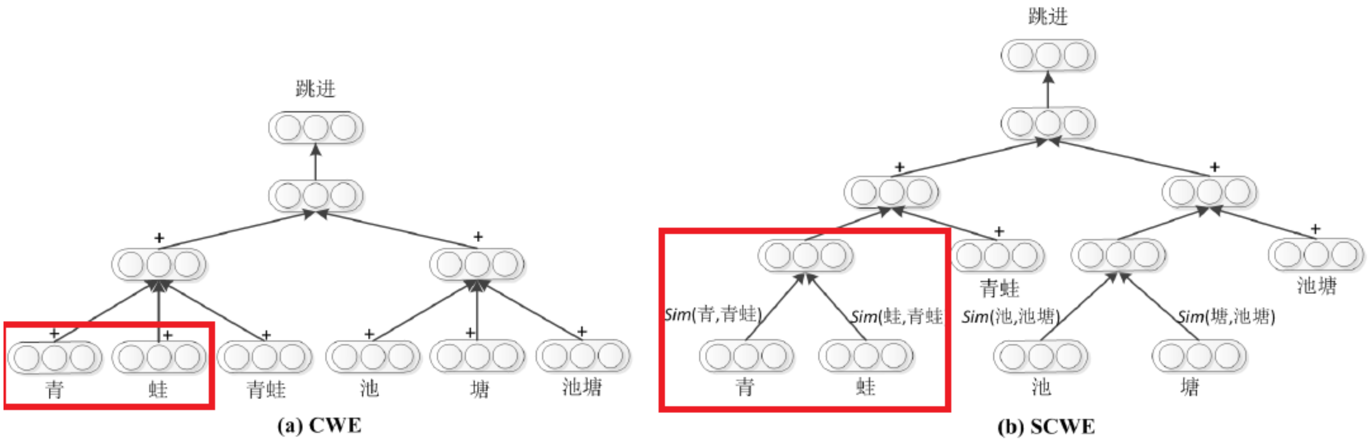
22-NAACL16-USTC-Improve ChineseWord Embeddings by Exploiting Internal Structure

PDF, Bib, CWE, SCWE

- [**Motivation**]

  - (Problem) Semantic Similarity across component characters in a word was ignored.
  - (One solution, maybe others solution) Learn semantic contribution of characters to a word **via** the semantic knowledge obtained from other languages.

  - (Some existing work) motivated by Chen et al.(2015)^1 explots the internal structures of Chinese characters.
  - (Our work) Chen et al.(2015) treat each character's equal contrubution to a word, while this work treat differently.



(a) CWE

(b) SCWE

- [**Contribution**]
  - 1. (Provide a method) Recongnize semenatically conpositional chinese word.
  - 2. (Provide a method) How to calculate Sim(,)
  - 3. **Novel way** to disambiguate Chinese characters with translating resources.
- [**Methodology**]
  - 1. Obtain tranlations of Chinese words and characters
  - 2. Perform Chinese character sense disambiguation
  - 3. Learn word and character embedding with our model
- [**Experiment**]
  - 1. Word Similarity (**SemEval2017 - Task2**)
  - 2. Text Classification
  - 3.1 Does this work solve ambiguilty of Chinese characters?
  - 3.2 Does this work make semantic effect?
  - 3.3 How does the parameter effect the performance?

  [^1] Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015. Joint learning of character and word embeddings. In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI).

12-NIPS13-Mikolov-Distributed Representations of Words and Phrases and their Compositionality

PDF, Bib

ICLR13-Mikolov-Efficient estimation of word representations in vector space, Bib

- [**Problem**] the cost of computing the gradient of the Skip-gram model is proportion to the vocabulary size.
- [**Method**]
  - **Hierachical Softmax**
    - Build a binary Huffman tree as the representation of the output layer with W words as its leaves, and for each non-leaves, explicitly represents the relative probabilities of its chiald nodes.
    - reduce to log(W)
  - **Negative Sampling**
    - Assumption: **A good model should be able to differentiate data from noise by means of logistic regression**.
    - replace the objective of nagative sampling, to distingush the target word from draws from the noise distrubution
- [**Others**]
  - **SubSampling of Frequent Words**
    - the most frequent words usually provide less information value than the rare words.
    - discarded with a probability
  - **Phrases Vector**
    - replace words to phrases
    - how to extract phrases? -- words that appear frequently together, and infrequently in other contexts.

This two paper proposed the skip-gram model, and tries to solve two aspects problems: 1). How to make it computable? 2). How to make it more semantical?

As to the first problem, the author tries two method, the one is to replace the softmax as hierachical softmax, which reduce the time complexity to log(W), and the other is to replace the objective with negative sampling.

And when it comes to the second problem, the author tries some tricks. Firstly, he subsampling the frequent words since the vector representations of frequent words do not change significantly. Secondly, he treat phrases as a kind of word to train phrase vectoe, because many phrases have a meaning that is not a simple composition of the meaning of the its individual words.

How to build a model? I think this papers pointed out the right directions.