0. Embeddings - информативное векторное представление объектов

Задачи NLP

- Sentiment analysis
- Spam filtering
- Fake news detection
- Topic prediction
- #hashtag prediction

} text classification problems

1) bag of words - каждое предл/текст - это бинарн. маска вокабуляра

Препроцессинг данных

- токенизация: разбиение на части (на слова/слоги/буквы)
- нормализация:
  - стемминг - отрубка до корня слова (примерно) (пример: портер, ланкастер, портер 2)
  - лемматизация - отрубаем до 1-ой формы слова для каждой части речи.
    (пример можно посмотреть в nltk)
  - удаление цифр, стоп-слов и т.д.

TF-IDF

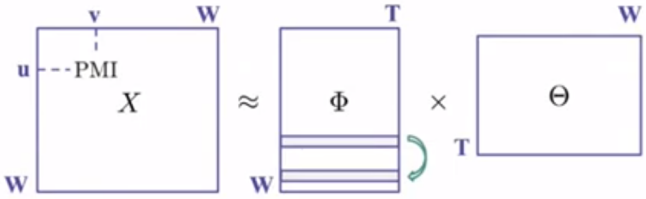
tf - term frequency - частота слова в документе

idf - inverse document frequency - логарифм отношение кол-ва док-тов к числу док-тов, где встречается это слово

1)

Word representations via matrix factorization

- **Input: PMI, word coocurrences, etc.**
- **Method: dimensionality reduction (SVD)**
- **Output: word similarities**



37

ı: Pointwise Mutual Information (PMI)

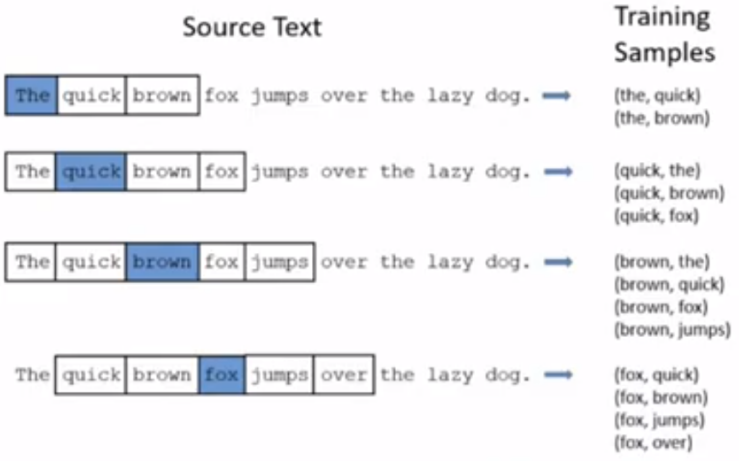
$$PMI = log\frac{p(u,v)}{p(u)p(v)} = log\frac{n_{uv}n}{n_u n_v}$$

Much better solution: **Positive PMI (pPMI)**

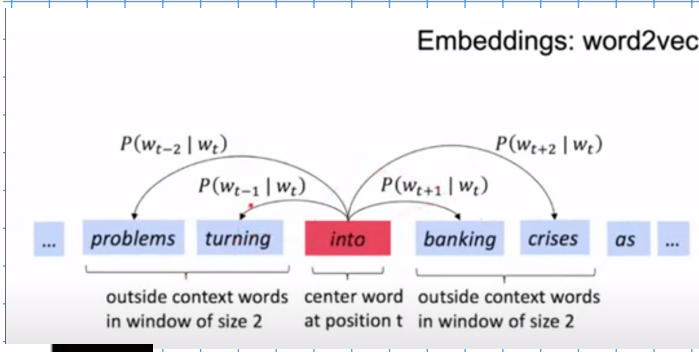
$$pPMI = \max(0, PMI)$$

Word 2 vec

Embeddings: word2vec

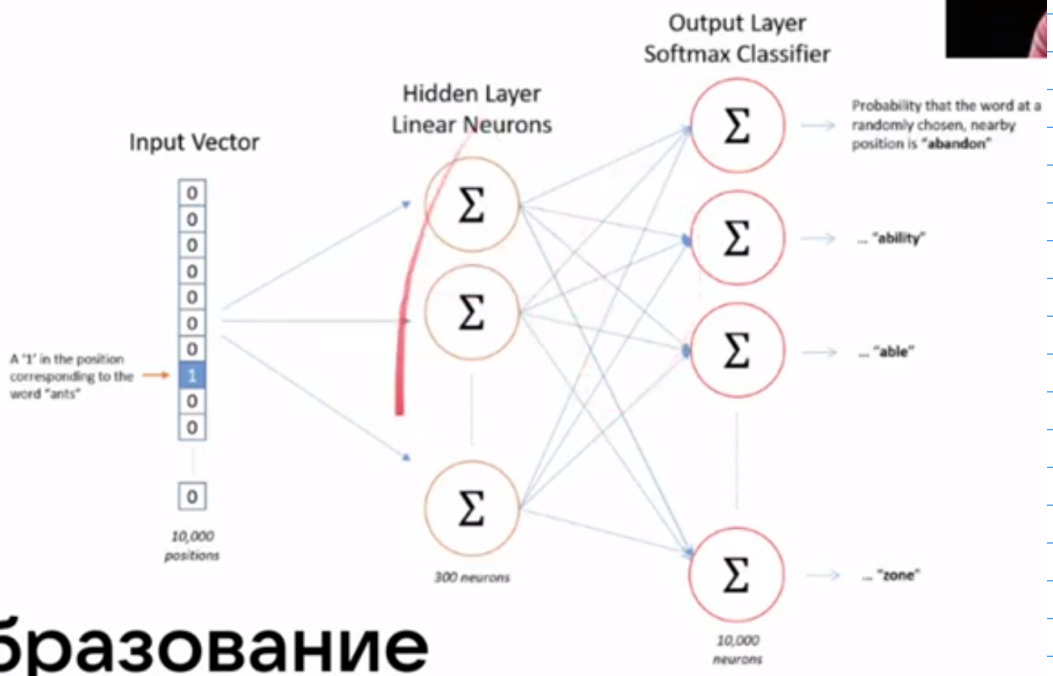


Embeddings: word2vec



45

Embeddin



образование

Embeddings: word2vec

Subsampling frequent words.

w_i is the word, $z(w_i)$ is the fraction of this word in the whole text

Graph for (sqrt(x/0.001)+1)*0.001/x

$P(w_i)$ is the probability of *keeping* the word:

$$P(w_i) = (\sqrt{\frac{z(w_i)}{0.001}} + 1) \cdot \frac{0.001}{z(w_i)}$$

Source: http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/

52

обучаемся чаще на редко встречаемых словах и реже на часто встречаемых

Embeddings: negative sampling

Negative Sampling idea: only few words error is computed. All other words have zero error, so no updates by the backprop mechanism.

More frequent words are selected to be negative samples more often.The probability for a selecting a word is just it's weight divided by the sum of weights for all words.

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^{n} (f(w_j)^{3/4})}$$

53

1 из решений проблемы дисбаланса класса

Continuous BOW (CBOW)

$p(w_i | w_{i-h}, \ldots, w_{i+h})$

Predict center word from (bag of) context words

- Predicting one word each time
- Relatively fast

Skip-gram

$p(w_{i-h}, \ldots w_{i+h} | w_i)$

Predict context ("outside") words (position independent) given center word

- Predicting context by one word
- Much slower
- Better with infrequent words