Word Embeddings (PA153)

Pavel Rychlý

Continuous space representation

- words represented by a vector of numbers
- similar words are *closer* each other
- more dimensions = more features
    - tens to hundreds, up to 1000

$$\textit{continue} = [0.286, 0.792, -0.177, -0.107, 0.109, -0.542, 0.349]$$

Simple vector learning

- each word has two vectors
  - node vector ($node_w$)
  - context vector (ctx_w)
- generate $(node, context)$ pairs from text
  - for example from bigrams: $w1$, $w2$
  - $w1$ is $context$, $w2$ is $node$
- move closer ctx_{w1} and $node_{w2}$

Word2vec

- command line tool for creating word embeddings
- two models:
  - CWOB = Continuous back of words
  - SKIP-GRAM
- many parameters
  - window size
  - dimension of vectors
  - alpha (learning rate)
  - min-count for words
  - sub-sampling limit

Word2vec

- simple tokenization = space separated
- lines = paragraphs (never crossed by window)
- negative sampling
- sub-sampling
- fast computation on multiple CPU
- compact, cryptic C

GloVe

- several (independent) modules
- clean C
- can save both node and context vectors

FastText

- includes character n-grams
- handling of unknown words, low-frequent words
- tangled, many-class C++
- many pre-trained models