Word embeddings

capture

semantical / morphological / hirearchial. info

Map words → real no. vectors

↓ reflects

structure, morphology, Content, hirearchy, relationships.

LSA (Latent Semantic Analysis)

- → low-dim data
- → analysis of incoming docs from same domain zone.
- → put into (term doc matrix)

← Code snippet →

One-hot encoding. (dis→ sparse matrix)

- → count-occurance. (count Vectorizer)
- → Matrix $\overset{\text{words}}{\left[\quad\right]}\Big\}$ docs

← Code snippet →

TF-IDF (term frequency, inverse doc freq)

- statistical weighting (tf-idf)
- stopwords filtering

(why → false claim of attaching importance to these words).

→ tf-idf soln for stopwords prblm:

$$\text{tf-idf}(\text{term}, \text{doc}) = \underbrace{\text{tf}(\text{term}, \text{doc})}_{} * \underbrace{\text{idf}(\text{term})}_{}$$

times the word occurs in doc / # total words in doc

inversed no. of docs in which the term we're interessted in occurs.

$$tf(term, document) = \frac{n_i}{\sum_{k=1}^{V} n_k}$$

$$idf(term) = \log \frac{N}{n_t}$$

Replacing each cell of it by tf-idf score

← Code snippet →

Word2Vec (parameter learning) → [child Doc2Vec] (pharagrap reptn).

- Neural embedding
- 3 types
  - One-word count
  - Multi-word count
  - Skip-gram model

One-word count: (CBOW model)
↳ one-word per one-content

Architecture: i/p → one-hot encoded vector → hidden layer weight matrix → o/p layer (softmax fn) (Activation step)

Goal → cal $P(w_j | w_I)$ probability dist. with. (vector rep. of word index I)

Activation step → calculated with standard softmax.

* (-ve sampling o/r hierarchical softmax ?)

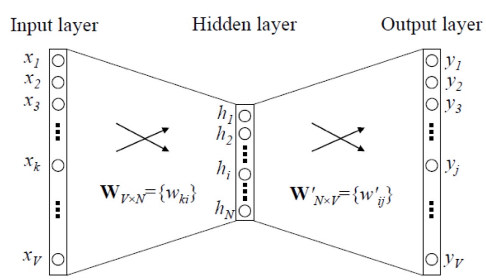

Figure 1: A simple CBOW model with only one word in the context

Multi-word content:
↳ same as above except in prob. distr & type of hidden layers.
↳ predicts multi-nomial distrbn given many word contexts. (stores relatn of target word to other words in corpus).

$P(w_o | w_{1,1} \ldots w_{i,c})$

Cost fn: $-\log P(w_o | w_{1,1} \ldots w_{i,c})$

Skip-gram model.
- opposite to CBOW multi-word model.
- predict (c) context words having one target on i/p

⟵ code snippet (Word2Vec) ⟶

GloVe (Global vectors for Word representn)

- goal: capture meaning of one-word embedding with structure of whole observed corpus.

word freq & co-occurance counts (measures)

Trains on global co-occurance counts, minimizes least-square error $\xrightarrow{\text{produces}}$ word vector space with meaningful substructures.

* Preserves word similarities with vector distances.
* Stores in Co-occurance matrix (X)

each entry = # times word j occurs in context of word (i)

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}$$ prob of word with index (j) occurs in context of word (i)

FastText (Enriches word vectors with sub-word info)

- considers internal structure of words by splitting them into bag-of-character n-grams & adding them to whole word as final feature.

Poincare embeddings

- latest in NLP
- uses hyperbolic geometry to capture hierarchical properties of words which we can't capture directly in Euclidean space.

Conclusions:

Real-life appln of these depend on task & given corpus.