word2vec - Word embeddings

Babak Maser, Hugo Platzer

October 23, 2018

Introduction: Natural language processing

The processing of natural language is important for many applications, including the following:

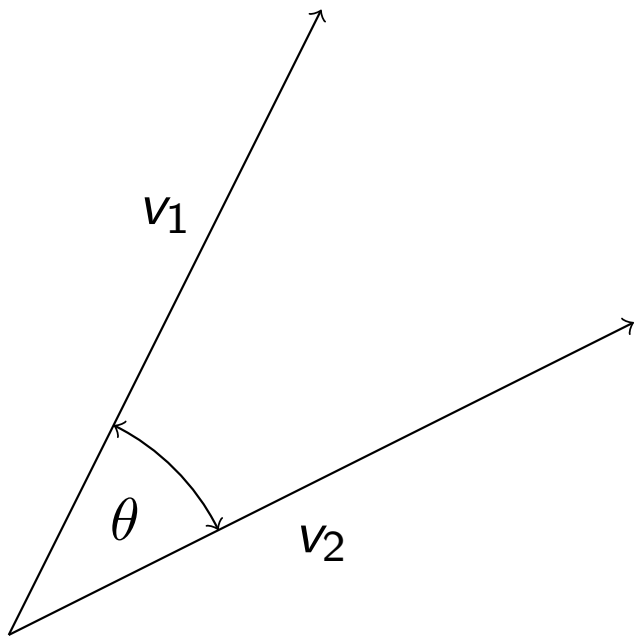
- Translation
- Sentiment analysis
- Web search
- Language modeling

Natural language processing (NLP) can analyze language

- **syntactically**: Parse grammar, check spelling etc.
- **semantically**: Understand meaning, find synonyms etc.

Vector space model

In this model of information retrieval for NLP, documents are represented by a vector of the counts of terms occurring in them. Terms can be words, keywords or phrases. Vector operations can now be used to compare documents for similarity:



$$\cos\theta = \frac{v_1 \cdot v_2}{||v_1||\ ||v_2||}$$

Vector Representation of word

one-hot representation:

The dimension of each word would be the number of unique words in the corpus.

$$\text{Book} = [\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ ]$$

Co-occurance Matrix:

Silence is the language of God, all else is poor translation. (Rumi 1207 , 1273)

	Silence	is	language	God
Silence	0	1	0	0
is	1	0	0	0
language	0	1	0	0
God	0	0	1	0

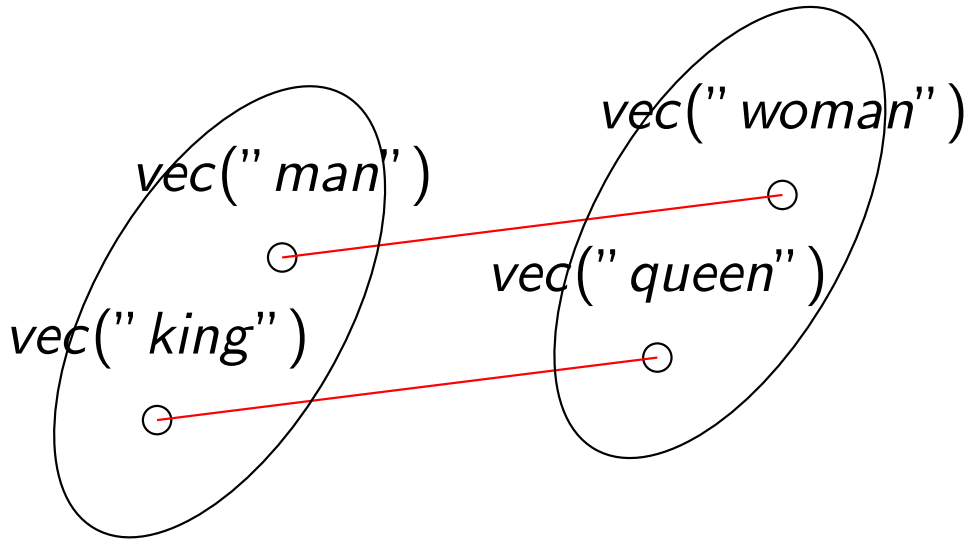
Word2vec

Word2vec is a language model that learns about the relationship between words. The output vectors show interesting relationships, e. g:

$$vec(\text{"}king\text{"}) - vec(\text{"}man\text{"}) + vec(\text{"}woman\text{"}) \approx vec(\text{"}queen\text{"}).$$

The model should put words that occur in similar context into clusters:

Word2vec



$$dist(vec(\text{"}king\text{"}), vec(\text{"}man\text{"})) \approx$$
$$dist(vec(\text{"}queen\text{"}), vec(\text{"}woman\text{"}))$$
$$\Rightarrow vec(\text{"}king\text{"}) - vec(\text{"}man\text{"}) \approx vec(\text{"}queen\text{"}) - vec(\text{"}woman\text{"})$$
$$\Rightarrow vec(\text{"}king\text{"}) - vec(\text{"}man\text{"}) + vec(\text{"}woman\text{"}) \approx vec(\text{"}queen\text{"}).$$

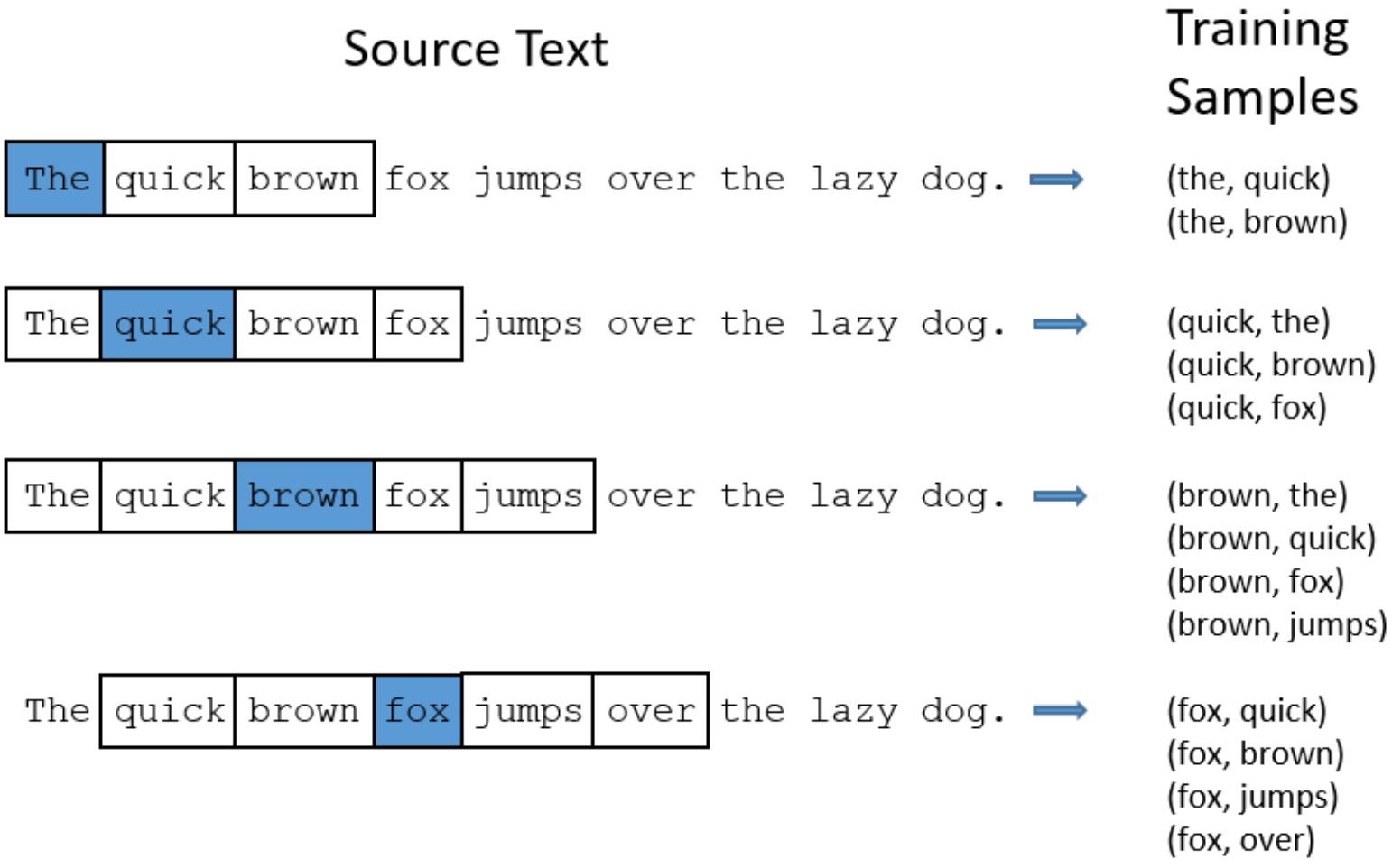
Word2vec

Methods :

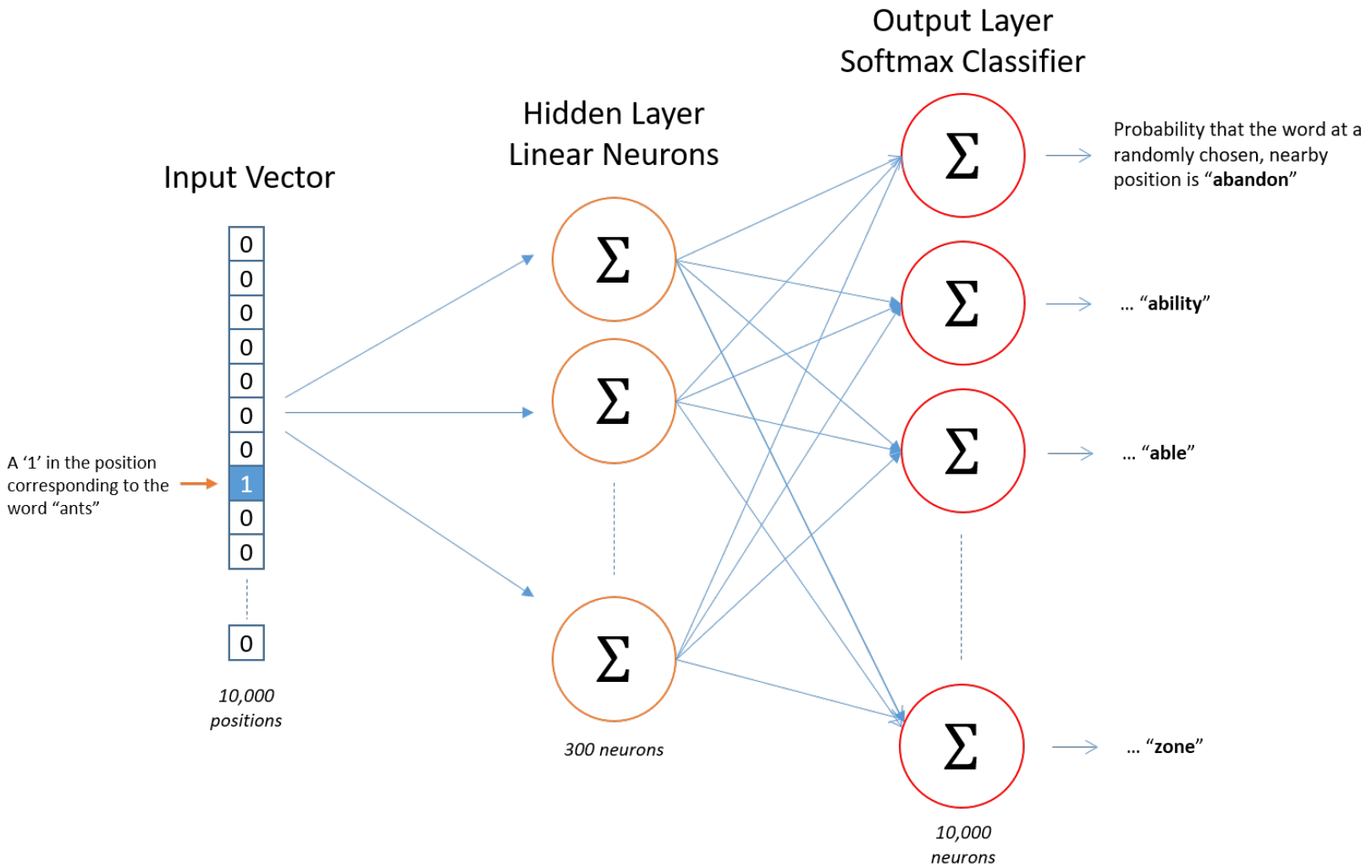
- CBOW (Continuous Bag of words)
- Skip-Gram

The CBOW architecture predicts the current word based on the context.
The skip-Gram predicts surrounding words given the current words.

The skip-gram model



The skip-gram model



The continuous-bag-of-words model

- The skip-gram model tries to guess context words based on the current word, the CBOW model goes the other way
- The one-hot vectors of the surrounding words are summed up (counting the context words), the order of the words is not considered, this is the network input
- The current word's one-hot vector is the desired output

Next steps

- Choose a minimal, but interesting corpus of English words
- Select a source of text to train the network from
  - Browse through Wikipedia or similar websites
  - Extract long enough passages only consisting of our limited vocabulary
- Generate skip-gram or CBOW pairs for network training
- Desired size of word vector / hidden layer size
- Potential problems with network training (large number of weights)
- Use test data to check prediction accuracy, adjust training parameters if necessary