POLIMI GRADUATE SCHOOL OF MANAGEMENT

INTRODUCTION TO NATURAL LANGUAGE PROCESSING

Andrea Mor - andrea.mor@polimi.it

CONTENTS

- Introduction
- Data Preparation
- Topic Modeling
- Sentiment Analysis
- Word embedding: Word2vec and Glove
- RNN - Long Short Term Memory networks (LSTM)

SOME IMPORTANT QUESTIONS

- What is the meaning of a word?
- How we can represent it?
- Which are the limits/problems of a representation?

TEXT CLEANING

- Convert to lower case
- Remove punctuation
- Remove numerical values
- Typos
- Remove special characters ([?@)
- Remove stop words (the, it, etc)
- Remove special description words ([chorus], [fade], [applause])
- Tokenize text (O'Neill $\rightarrow$ [o] [neill], [o'neill]?; aren't $\rightarrow$ [arent], [are][nt]?)
- Create bi-grams or tri-grams ([United Kingdom] vs [United][Kingdom])
- Normalization:
  - Stemming (car, cars, car's, cars' $\rightarrow$ car;)
  - Lemmatization (am, are, is $\rightarrow$ be)

TEXT REPRESENTATION

1. Corpus: a collection of text
2. Document-Term Matrix: word counts in matrix format
3. TF-IDF: Term Frequency - Inverse Document Frequency

$$\text{TF-IDF} = f_{t,d} \times \text{idf}(t, D)$$

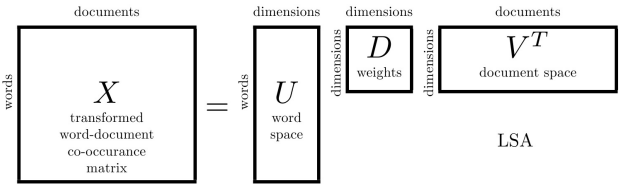
where

- $tf(t)$: the number of times that term t occurs in document d.
- $\text{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}| + 1}$: log of the inverse of the number of documents containing term t.

SENTIMENT ANALYSIS

- TextBlob Module: Linguistic labeled the sentiment of words. https://github.com/sloria/TextBlob/blob/eb08c120d364e908646731d60b4e4c6c1712ff63/textblob/en/en-sentiment.xml
- Sentiment Labels: Each word is labeled in terms of
  - Polarity: negative(-1) or positive(+1)
  - Subjectivity: subjective(0) or fact(+1)
- Sentiment of words can vary based on where it is in a sentence.
  - Negation multiplies the polarity by -0.5

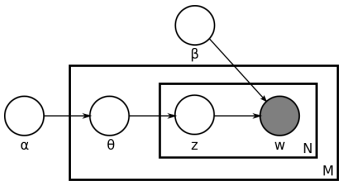
TOPIC MODELING - LSA



Latent Semantic Analysis (LSA)

- **Singular Value Decomposition (SVD)** of the Document-Term Matrix

TOPIC MODELING - LDA



Latent Dirichlet Allocation (LDA)

- **Documents (M) are probabilistic distribution over topics**: let say that a document is p_i% of topic i.
- **Topics are probabilistic distribution over words**: given a topic chosen according to the distribution of the document, we generate a word according to the topic distribution
- Random initialisation: assign each word to a random topic
- Update each word by considering
  - proportion of words in the document of topic
  - proportion of topics in all documents for the word
- We sample until a “reasonable” result

WORD EMBEDDING

Distributed Representations of Words (a.k.a. word embeddings) are geometric representation of words/entities learned from the data/corpus in such a way that semantically related words are often close to each other.

In practice we attempt to embed entities onto a low -dimensional metric space in which similar words are placed close

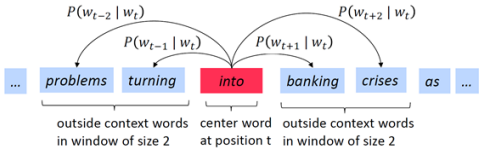
DISTRIBUTIONAL HYPOTHESIS

“You shall know a word by the company it keeps”

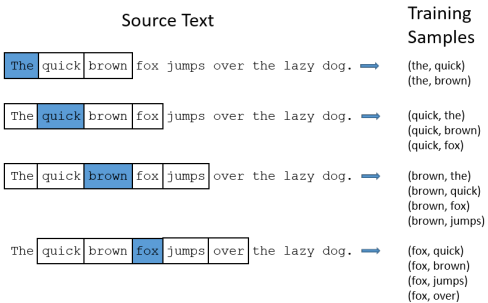
(Firth, 1957)

Similar words tend to occur in similar contexts, therefore a word can be represented based on the co-occurrence across the data in the same context (word window).

WORD2VEC INPUT

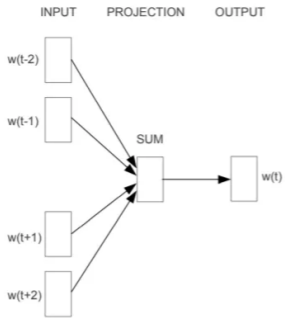


1

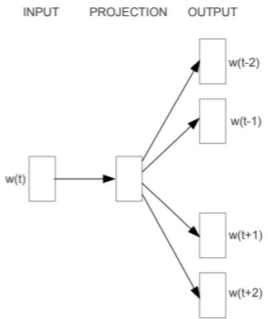


1 http://web.stanford.edu/class/cs224n/

WORD2VEC: CBOW VS SKIP-GRAM



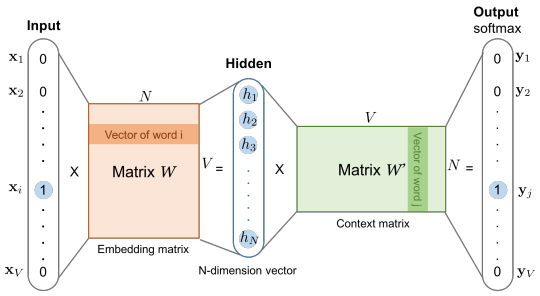
CBOW



Skip-gram

WORD2VEC SKIP-GRAM

A single-layer architecture based on the inner product between two word vectors.

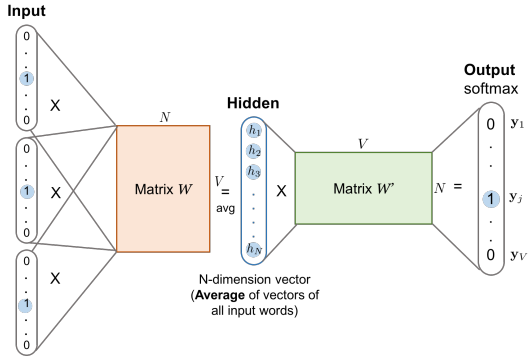


$$\underbrace{e_i^\top \times W_{N\times d}}_{h} \times W'_{d\times N} \rightarrow^{\text{softmax}} \mathbb{P}(\text{word}_j|\text{word}_i)$$

We maximize $\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq k\leq c,k\neq 0} log(p(w_{t+k}|w_t))$ where $p(o|c) = exp(v_o^\top v_c)/\sum_{w=1}^{V} exp(u_w^\top v_C)$

https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html

WORD2VEC CBOW



$$h = \frac{1}{|\text{window}|} \sum_{i=1}^{|\text{window}|} e_i^\top \times W_{N \times d}$$

https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html

GLOVE - GLOBAL VECTORS

- We use the co-occurrence matrix for the entire corpus.
- X_{ij}: # of times word j is in the context of word i.
- $P(j|i) = \frac{X_{ij}}{X_i}$ with $X_i = \sum_l X_{il}$, probability that j is in the context of i

Probability and Ratio	k = *solid*	k = *gas*	k = *water*	k = *fashion*
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

COSINE SIMILARITY

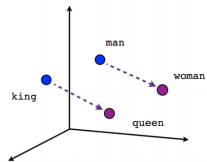
Similarity distance measure as the cosine similarity:

$$similarity(a, b) = \frac{a^\top b}{||a||\,||b||} = cos(\theta)$$

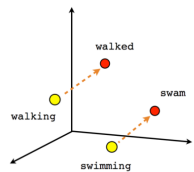
ANALOGICAL REASONING

The city of Rome is in relation with the country Italy in the same way as the city of Paris is in relation with the country France.

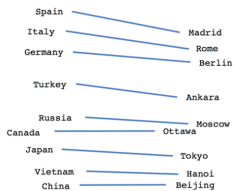
The propositional analogy task: find an x such that Rome : Italy = x : France



Male-Female



Verb tense



Country-Capital

$$v(Italy) - v(Rome) \sim v(France) - v(Paris)$$

IMPLEMENTATIONS

- Neural Network: Word2Vec [Mikolov+, 2013], ELMO [Peters+,2018]
  https://code.google.com/archive/p/word2vec
- GloVe [Pennington+,2014]. Matrix Factorization/Neural Network.
  https://nlp.stanford.edu/projects/glove/

THANK YOU