Lehrstuhl für Datenverarbeitung
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

Distributed Representations of Words and Phrases and their Compositionality

Authors of the paper: Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean

Student poster created by: Felix Buchert, Daniela Geisinger, Johann Jahner, Alexander Ladwein, Beike Lu, Alexander Schwer

1. Motivation

- Basis: Skip-gram neural network model using vector representations for words
- Goal: Improving the quality of the vectors and the training speed
- Approach: Extending the Skip-gram model with subsampling of frequent words, negative sampling, as well as the learning of idiomatic phrases

2. The Skip-gram Model

- Goal: Find **word representations** that are useful for predicting the surrounding words, the context, given an input word.
- Approach: Training objective is to maximize the average log of the probability

$$p(w_O \mid w_I) = \frac{\exp({v'_{wo}}^T v_{WI})}{\sum_{w=1}^{W} \exp({v'_w}^T v_{WI})}$$

- Output layer: Softmax layer that computes the probabilities for each word in the vocabulary to be inside the context
- Limitations: $\boldsymbol{cost\ of\ computation} \propto \boldsymbol{W}$, the number of words in the vocabulary

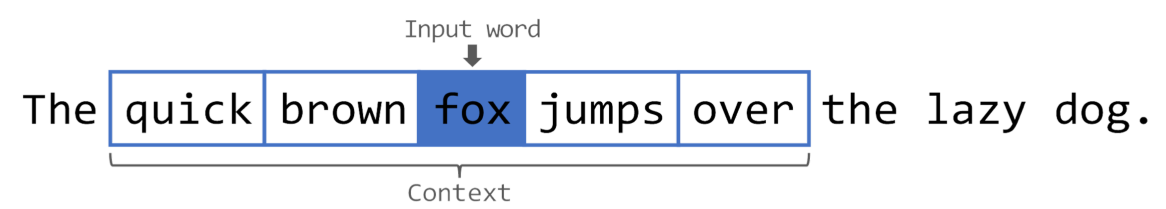

Figure 1: Example of the context for a given input word and a window size of two.

2.1 Hierarchical Softmax (HS)

- Goal: Increase the computational efficiency
- Approach: Replaces the flat Softmax output layer with a **binary tree** layer. Each leaf represents a word in the vocabulary and the product of all node probabilities on the path from the root to the leaf represents the probability $p(w_O|w_I)$ which is defined as follows

$$p(w \mid w_I) = \prod_{j=1}^{L(w)-1} \sigma\left(\left[n(w, j+1) = ch\left(n(w,j)\right)\right] * {v'_{n(w,j)}}^T v_{wI}\right)$$

- Tree Structure: **Binary Huffman tree** is used for fast training; it uses short codes for frequent words
- Increase in efficiency with HS: $\boldsymbol{cost\ of\ computation} \propto \log(\boldsymbol{W})$

2.2 Negative Sampling (NEG)

- Goal: Computationally inexpensive objective for learning word vector representations
- Alternative to Hierarchical Softmax: The Negative Sampling objective

$$\log\left({v'_{w_O}}^T v_{w_I}\right) + \sum_{i=1}^{k} E_{w_i \sim P_n(w)} \left[\log \sigma\left(-{v'_{w_i}}^T v_{w_I}\right)\right]$$

replaces every $\log P(w_O|w_I)$ term in the Skip-gram model

- Task: Distinguish the target word w_O from draws from the noise distribution $P_n(w)$ using logistic regression
- Negative Samples: For every data sample k negative samples are drawn from the noise distribution
- Noise distribution $P_n(w)$: In general the choice of the distribution is free, but the unigram distribution raised to the 3/4rd power has empirically found to significantly outperform other choices

2.3 Subsampling of Frequent Words

- Motivation: Vector representations of frequent words do not change significantly after training on large datasets, as they usually provide less information than rare words
- Goal: Approach that efficiently discards frequent words while preserving the accuracy of the learned vectors
- Approach: Each word w_i is discarded with probability

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \ ,$$

with frequency $f(w_i)$ of word w_i and threshold t (around 10^{-5})

2.4 Results

- Evaluation: The described methods were evaluated on the basis of the analogical reasoning task considering syntactic and semantic analogies
- Training parameters: Vocabulary size of 692K, Dimension size of 300

Method	Time [min]	Syntactic [%]	Semantic [%]	Total accuracy [%]
NEG-5	38	63	54	59
NEG-15	97	63	58	61
HS-Huffman	41	53	40	47
The following results use 10^{-5} subsampling				
NEG-5	14	61	58	60
NEG-15	36	61	61	61
HS-Huffman	21	52	59	55

Table 1: Computation time nd accuracy of various Skip-gram models on the analogical reasoning task. NEG-k, with k denotng the number of negative samples for each positive sample

3. Learning Phrases

- Motivation: Meaning of phrases is not given by the mere composition of its words
- Approach: Find a single representation for every phrase by detecting words that appear frequently together and rarely in other context
- Result: Training with different Skip Gram Models (hyperparameter) shows that subsampling improves performance also significantly for phrases
- Size of training data: By evaluating the results of analogy task, it was found that the amount of data is crucial (33B→72%; 6B→66%)

4. Additive Compositionality

- Linearity: due to the linear structure of the representation, also vector addition gives reasonable output (distr. are multiplied → AND-function)
- e.g.: vec("Berlin") – vec("Germany") + vec("France") = vec("Paris")

5. Conclusion

- the model architecture and the subsampling of frequent words both result in a computationally more efficient training and improve the quality of the word representations, in particular for rare and uncommon words, respectively
- the performance is sensitive towards both the training algorithm and the hyperparameters, which is why they should be selected depending on the specific task