Peepoo Robot Language

BSc Omar Mayani

MSc thesis
June 2026

MATEMATIIKAN JA TILASTOTIETEEN LAITOS

Reviewers:
Prof. H. H.
PhD D.D.

UNIVERSITY OF TURKU, Department of Mathematics and Statistics

MSc Thesis
Subject: Mathematics
Author: Omar Mayani
Title: Peepoo Robot Language
Supervisor: Prof. H. H.
Pages: xx pages + xx appendix pages
Month and year: June 2026

Write the abstract here. Insert the \noindent instruction in the beginning of the paragraph to prevent the first line indention in LaTeX.

Command \vspace leaves space between the paragraphs, 4mm looks good.

Write the abstract compactly, avoiding unnecessary repetition.

Keywords: Abstract, MSc thesis, LaTeX system.

ON NOTATION AND TERMINOLOGY

Unless otherwise specified, we use the following notations and terminologies throughout the thesis.

$\mathbb{R}$	the set of real numbers,
L	the loss function,
T	the transpose of a matrix,
w	a member of a vocabulary (usually a word),
$V = \{w_1, w_2, \ldots, w_{\lvert V \rvert}\}$	a (finite) vocabulary,
$\mathbf{w}$	the vector representation of w,
$\mathbf{w}(w)$	the vector representation of w,

Contents

1 Introduction

This section includes general despription of the field and the topic of the MSc thesis.

Note that if you have used AI in your writing, you must mention it. Check for guidelines https://utuguides.fi/artificialintelligence. Good place to mention that AI tolls have been used is here, for example, in the following form:

The ChatGPT AI has been used to enhance the language of the thesis.

2 Foundations of Statistical Learning for NLP

Theorem 1. *Assume that in interval* $[a, b]$ $F'(x) = f(x)$ *holds and that* f *is continuous.*[1] *Then*

$$\int_a^b f(x)\,dx = F(b) - F(a).$$

[1] The continuity is essential here.

3 Representation Learning

Machines need to understand each word first so as to master the sophisticated meanings of human languages. Hence, effective word representations are essential for natural language processing (NLP), and it is also a good start for introducing representation learning in NLP.

The form of a word representation can be divided into two categories: symbolic and distributed representations. In symbolic representation, each word is represented as a unique symbol or index, such as one-hot encoding. In distributed representation, each word is represented as a dense vector in a continuous vector space, where semantically similar words are mapped to nearby points.

Symbolic representations are easy to implement and interpret, but they suffer from the curse of dimensionality and cannot capture semantic relationships between words. The distributed word representation overcomes these problems by representing words as low-dimensional real-valued dense vectors.

In distributed word representation, each dimension in isolation is meaningless because semantics is distributed over all dimensions of the vector. Distributed representation can be obtained by factorizing the matrices of symbolic representations, such as in Latent Semantic Analysis LSA, or by optimizing the word vectors with gradient descent to predict the context words, such as in Word2Vec. Glove is a hybrid method that factorizes the co-occurrence matrix of words while also optimizing the word vectors to predict the co-occurrence counts.

3.1 Symbolic Representations

One-hot

$$\mathbf{w_j^{(i)}} = \begin{cases} 1, & \text{if } j = i, \\ 0, & \text{otherwise,} \end{cases}$$

Given a vocabulary $\mathcal{V}$ of size n and a corpus $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ of m documents, the Bag of Words (BoW) representation of this data is the word–document matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$ where $\mathbf{M}_{i,j}$ is the count of word w_i in document d_j. Often one would also scale these counts by the length of the document so that $\mathbf{M}_{i,j} = \frac{\text{count of } w_i \text{ in } d_j}{|d_j|}$.

While frequency captures how often a word appears, it fails to capture how **informative** a word is. Common words such as "the" and "is" will be frequent across all documents but no document is distinguished by the presence of these words. To address this we multiply the term frequency (tf) by the inverse document frequency (idf).

The term frequency of a word w_i in document d_j is defined as

$$\text{tf}(w_i, d_j) = \frac{\text{count of } w_i \text{ in } d_j}{|d_j|}.$$

The inverse document frequency of a word w_i is defined as

$$\text{idf}(w_i) = \log\left(\frac{|\mathcal{D}|}{|\{d \in \mathcal{D} : w_i \in d\}|}\right)$$

3.2 Distributed Word Representations

REWRITE Although simple and interpretable, symbolic representations are not the best choice for computation. For example, the very sparse nature of the symbolic representation makes it difficult to compute word-to-word similarities.

The difficulty of symbolic representation is solved by the distributed representation. Distributed representation represents a subject (a word in our case) as a fixed-length real-valued vector, where no clear meaning is assigned to any single dimension of the vector. More specifically, semantics is scattered over all (or a large portion) of the dimensions of the representation, and one dimension contributes to the semantics of all (or a large proportion) of the words

3.2.1 Matrix Factorization-based Approaches

We briefly introduce three matrix factorization-based approaches for learning distributed word representations: Latent Semantic Analysis (LSA), Probabilistic LSA, and Latent Dirichlet Allocation (LDA).

Latent Semantic Analysis (LSA) Before the era of prediction-based neural models, the dominant paradigm for extracting semantic information from text was statistical count-based modeling. The most prominent of these methods is Latent Semantic Analysis (LSA), which rests on the hypothesis that words with similar meanings will occur in similar text segments. Mathematically, LSA is an application of low-rank matrix approximation using Singular Value Decomposition (SVD).

Let $\mathcal{V}$ be a vocabulary of size $|\mathcal{V}| = n$ and let $\mathcal{D}$ be a corpus of $|\mathcal{D}| = m$ documents. We construct a word-document matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$, where $\mathbf{M}_{i,j} = \text{tfidf}(w_i, d_j, \mathcal{D})$. We can then apply SVD to factorize $\mathbf{M}$ into three matrices

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\intercal,$$

where $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{m \times m}$ are orthonormal and $\Sigma \in \mathbb{R}^{n \times m}$ is a diagonal matrix containing singular values in descending order.

Since n and m are large preserving the full decomposition is computationally expensive and retains noise. We are motivated to take a low rank approximation by truncating the decomposition to the k largest singular values (with k being much smaller than both n and m)

$$\hat{\mathbf{M}} = \hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{V}}^\intercal,$$

where $\hat{\mathbf{U}} \in \mathbb{R}^{n \times k}$, $\hat{\mathbf{V}} \in \mathbb{R}^{m \times k}$ and $\hat{\Sigma} \in \mathbb{R}^{k \times k}$.

We can calculate the dot product similarity between two words w_i and w_j as

$$\begin{aligned}\mathbf{M}_i(\mathbf{M}_{:,j})^\intercal &\approx (\hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{V}}^\intercal)_i(\hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{V}}^\intercal)^\intercal_{:,j} \\ &= (\hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{V}}^\intercal)_i(\hat{\mathbf{V}}\hat{\Sigma}\hat{\mathbf{U}}^\intercal)_{:,j} = \hat{\mathbf{U}}_i\hat{\Sigma}^2(\hat{\mathbf{U}}_j)^\intercal.\end{aligned}$$

The above calculation shows us that the row vectors of $\hat{\mathbf{U}}\hat{\Sigma}$ (or $\hat{\mathbf{U}}$) can be used as word representations.

3.2.2 Neural Network-based Approaches

Neural network-based approaches have in recent years become the dominant method for learning distributed word representations. Instead of factorizing the symbolic representation, neural network-based approaches directly optimize the word vectors with gradient descent. The most popular neural network-based approach is Word2Vec, which was introduced by Mikolov et al. in 2013. Word2Vec is based on the distributional hypothesis, which states that words that occur in similar contexts tend to have similar meanings.

Word2Vec Word2Vec adopts the distributional hypothesis but does not take a count-based approach. It directly uses gradient descent to optimize the representations of a word toward tis neighbors' representations. There are two main architectures for Word2Vec: Continuous Bag of Words (CBOW) and Skip-Gram. The difference is that CBOW predicts the target word based on multiple context words, while Skip-Gram predicts the context words based on the center word.

Formally, CBOW predicts the word w_i as

$$P(w_i|w_{i-l},\dots,w_{i-1},w_{i+1},\dots,w_{i+l}) = \sigma\left(\mathbf{W}\sum_{\substack{j=i-l\\ j\neq i}}^{i+l}\mathbf{w}_j\right),$$

where $2l+1$ is the context size, σ is the softmax function, $\mathbf{W}\in\mathbb{R}^{|V|\times m}$ is the weight matrix and $\mathbf{w}_j$ is the embedding of word w_j.

The CBOW model is optimized by minimizing the sum of the negative log probabilities:

$$\mathcal{L} = -\sum_{i=1}^{N}\log P(w_i|w_{i-l},\dots,w_{i-1},w_{i+1},\dots,w_{i+l}).$$

Contrary to CBOW, Skip-Gram predicts the context words based on the center word. Formally, given a word w_i, Skip-Gram predicts the context words as

$$P(w_j|w_i) = \sigma\left(\mathbf{W}\mathbf{w}_i^{\intercal}\right), \quad j\in\{i-l,\dots,i-1,i+1,\dots,i+l\}.$$

GloVe

Appendix A Linear Algebra

blbalbal

Appendix B Probability theory

blbalbak

References

[1] Newton and Leibniz

[2] Riemann