# Delivery Report

Progetto RASTA, fase di Topic Modeling

## Contents

1. Topic Modeling 2

2. Word Representation 3

2.1. Local Representation 3

2.2. Distributional Embeddings 3

2.3. Contextual Embeddings 3

3. Document Representations 3

## List of Figures

[**Figure 1.1**: Overview of topic modeling. A corpus of documents is given as input and the model returns the list of topics and the topic representations of the documents. 2](#_Toc172574868)

## Topic Modeling

 Topic models are a class of models that provide an automatic way to analyze the main themes of large volumes of texts. A topic model describes a corpus of documents through a set of fixed topics, where each topic is represented by its most significant words. Figure 1.1 sketches how a topic model works, along with its input and outputs.

**Figure 1.1**: Overview of topic modeling. A corpus of documents is given as input and the model returns the list of topics and the topic representations of the documents.

A topic model presents the information in a compact and interpretable form. For example, as seen in the picture, a topic characterized by the words “*learning, machine, deep, neural, network”* can be easily interpreted as a topic related to deep learning, or the words “*probability, distribution, gaussian, variable, random*” are related to probability theory. However, a topic is not just an unordered list of keywords: each word of the vocabulary has a specific weight, or probability weight, that identifies the importance of the word in the topic.

Not only does a topic model summarize a corpus by lists of coherent keywords, but each document can be described by the discovered topics in different proportions. Indeed, a document is rarely characterized by a single topic, rather it may talk about multiple topics. For example, an NLP paper can contain 30% of linguistics and 70% of computer science.

The two main elements in topic modeling are the documents and its constituents, i.e. the words. To allow the topic model to deal with these elements, we need to find a way to represent them under a computational point of view. This Chapter is therefore organized as follows: we will provide an overview of the main representation methods for words and documents in Section 1.2. and Section 1.3. respectively. In this way, we will have the fundamentals to provide some details on the focus of this thesis, i.e. probabilistic topic models, in Section 1.4. Since these models are usually controlled by hyperparameters, we will also provide an overview of the main methodologies for estimating the hyperparameters in topic models in Section 1.5.

## Word Representation

Language is made of words that under a computational point of view come from a vocabulary and we need to find ways to account for the meaning of these words under a computational point of view. Nowadays, the most common way to represent words in NLP is to use **vectors in a vector space**: words are embedded in a multi-dimensional vector space and we can therefore interpret them as points in a space that can be compared. The process of “embedding” words in the vector space is what brought the community to call these representations *word embeddings*. "Word embeddings" is in fact the general term that is used to refer to this kind of representations.

We can distinguish between two different ways of representing words with vectors: we refer to the first as a **local representation** and to the second one as a **distributed representations** (Ferrone, 2017). This distinction derives from one of the most vivid debates in the AI field during the ’80s on how to store conceptual information inside neural algorithms. Local representations are meant to represent a single concept with the activity of a single neural unit. On the other hand, distributed representations are meant to account for a pattern of activity of more neural units (Hinton, 1989).

### Local Representation

The simplest way to represent words in a way that is interpretable from a machine consists in using **one-hot encoding**. In one-hot encoding, each word is represented by a single and unique vector. One-hot encoding maps the i-th word of the vocabulary *V* to the vector in a vector space , where n is the cardinality of the vocabulary *V* and the i-th element of is set to 1, while all the other elements are set to zero. We generally refer to this kind of vectors as the *one-hot vectors*.

For example, given the words of the vocabulary V = {the, cat, is, on, table}, the word “*cat*” of the vocabulary *V* can be represented as a vectors of zeros with only one 1 in the position indexed by its own index in the vocabulary = <0, 1, 0, 0, 0> (i.e., “*cat*” is the second element and thus the 1 will be in the second component of the vector).

If we want to represent all the unique words in a text corpus, we will then need a matrix whose dimension is *V x V* (where *V* is the number of unique words). A one-hot encoded representation is simple but results in two main issues. First, the dimension of the matrix grows as the number of unique words increases. Encoding all the words of the English vocabulary would generate a matrix of at least 170, 000 x 170, 000 entries. Related to this issue, the resulting matrix is extremely sparse (each row is composed of all 0-valued entries except for one entry). Each vector is orthogonal to each other, therefore not representing any type relationship between words. It is instead more convenient that word embeddings reflect and preserve certain specific properties of language. For example, we may agree that the word “*cat*" is more similar to "*dog*" than to “*Rome*". In the next section, we will see how this problem can be addressed by distributional representations.

### Distributional Embeddings

### Contextual Embeddings

## Document Representations

# Bibliography

Ferrone, L. a. (2017). Symbolic, distributed and distributional representations for natural language processing in the era of deep learning: a survey.

Hinton, G. E. (1989). *Parallel distributed processing: Explorations in the microstructure of cognition, chapter Distributed Representations* (Vol. vol.1).