Natural Language Processing

Mairi Hallman

June 2024

Contents

1 Introduction

Natural Language Processing (NLP) methods may either take in unstructured natural language as input, or produce it as output.

Much like the humans who use them, natural languages are as abiguous as they are variable. Consider the following examples:

I love cooking, my family, and my cat. I love cooking my family and my cat.

The addition of two commas completely changes the meaning of the sentence.

Humans excel at using language (expression, perception, and nuance), but fall short when it comes to understanding and explaining the axioms that govern language. Without these axioms, it is very difficult to utilize algorithms that require a more formal lingustic framework. As a consequence, computers typically have a hard time interpreting and outputting natural language.

Machine learning methods, particularily subervised learning algorithms, can be quite effective when defining "good" rules is a challenge. However, annotating outputs is simple compared to what humans can do (even if it is resource-intensive).

Challenge:

- input is variable and/or ambiguous
- rules are unknown and/or poorly defined
- natural languages are...
  - **discrete** Variables like colour or sound frequency can be expressed on continuous scales. How can you represent "blue" and "red" or "C sharp" and "A flat" on continuous scales? (Hint: you can't.)
  - **compositional** Characters say words, words make up sentences, and sentences make up stories. This is not a concept that can be easily explained in the form of an algorithm.
  - **sparse** Available natural language training data is an incredibly sparse set of the space of valid language statements. Many of the sentences you hear or read on a daily basis are likely sentences that have never been stated before. Language also evolves along with its users; for example, the word "slay" did not mean in 2005 what it means in 2025. "Skibidi" has only existed for a few years. You can't teach a computer words and sentences you've never met.
  - **symbolic** A word's meaning can't be inferred from its characters. Consider the words "pizza", "calzone", and "pizzaz". "Pizza" is much closer to "pizzaz" with respect to its characters, but much closer to "calzone" in terms of meaning.

2 Learning Basics and Linear Models

In supervised learning, we create mechanisms that attempt to generalize from labelled examples. In this section, we will use the words "model" and "function" interchangeably. Consider a set of emails labelled as "spam" or "not spam". A supervised learning model would attempt to generalize based on this training set to predict whether new emails are spam or not.

The space of all possible models is incredibly large, and treating this as an unconstrained problem would be computationally infeasible. We therefore limit the search by constraining the function to a specific family or **families**, such as linear functions or decision trees. The downside of constraining the hypothesis class is that in can lead to inductive bias. Introducing constraints leads to strong assumptions about the form of the solution. A constrained solution *may* turn out to be a valid approximation, or it may be way off. Finding a solution more quickly isn't helpful if the solution is innacurate.

Let $\mathbb{R}^{d_{in}}$, $\mathbb{R}^{d_{out}}$ represesent the spaces of **inputs** and **outputs**, respectively; then

$$f_\theta : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}} \qquad (W \in \mathbb{M}_{d_{in} \times d_{out}}(\mathbb{R}), \vec{b} \in \mathbb{R}^{d_{out}})$$
$$\vec{x} \mapsto \vec{x}W + \vec{b}$$

3 Working with Natural Language Data

4 Case Studies of NLP Features

5 From Textual Features to Inputs

6 From Textual Features to Inputs

7 Language Modelling

8 Pre-Trained Word Embeddings

9 Using Word Embeddings

10 Case Study of Sentence Meaning Inference