Foundations of Deep Learning

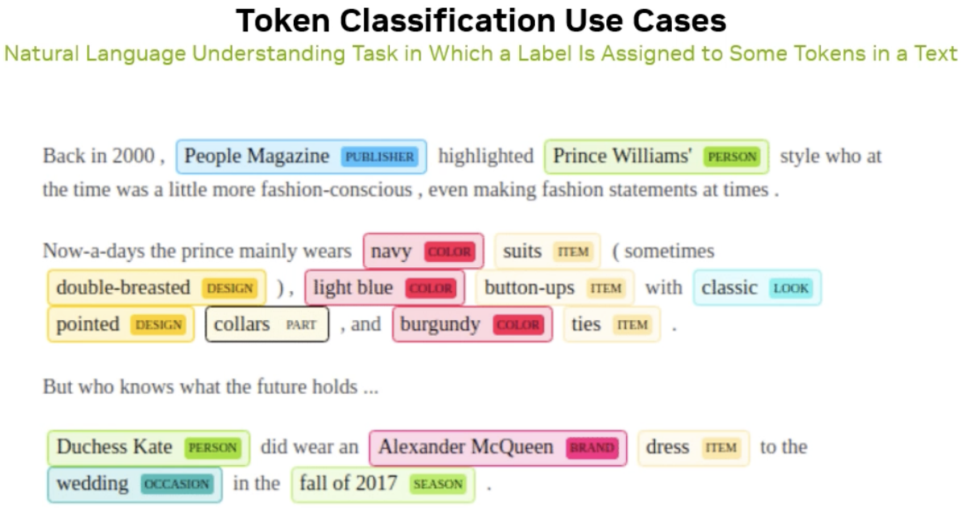
Geralyn Chong

March 17, 2025

Contents

1 Token classification

Token classification transforms unstructed into structured data. For example, Name-Entity recognition is a kind of token classification is used to recognize certain entities in a text by labeling segments of text:



Token Classification Use Cases

Natural Language Understanding Task in Which a Label Is Assigned to Some Tokens in a Text

Back in 2000 , People Magazine highlighted Prince Williams' style who at the time was a little more fashion-conscious , even making fashion statements at times .

Now-a-days the prince mainly wears navy suits (sometimes double-breasted) , light blue button-ups with classic pointed collars , and burgundy ties .

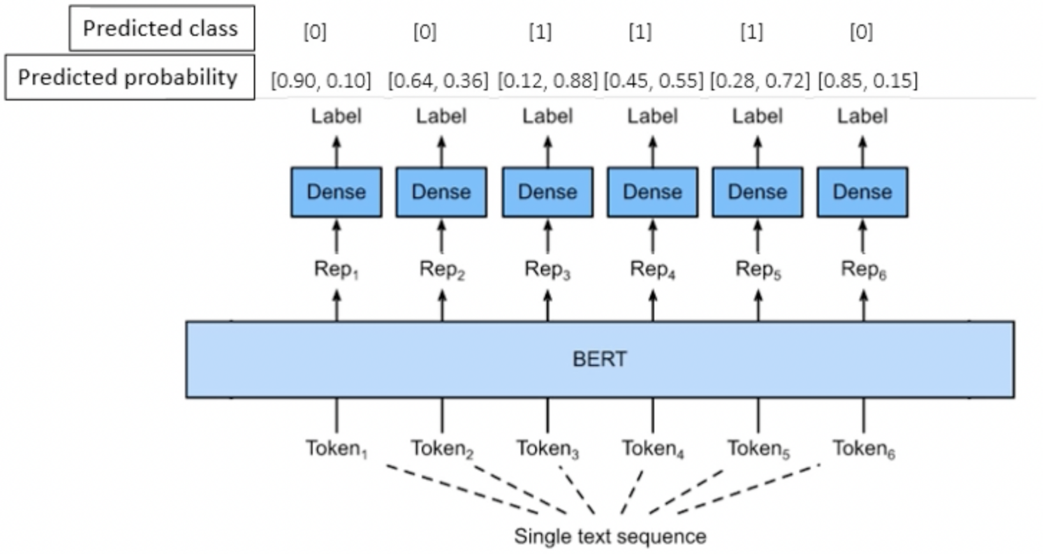
But who knows what the future holds ...

Duchess Kate did wear an Alexander McQueen dress to the wedding in the fall of 2017 .

We can also identify custom entities for specific task use cases. The example below indicates a type of tokenization architecture that includes BERT (Bidirectional Encoder Representations for Transformers) and a classification layer. In this case, we want to assign labels to the output from BERT. A hugging face tutorial that is adjacent to this concept.

Model Architecture

Language Model + Classification Layer

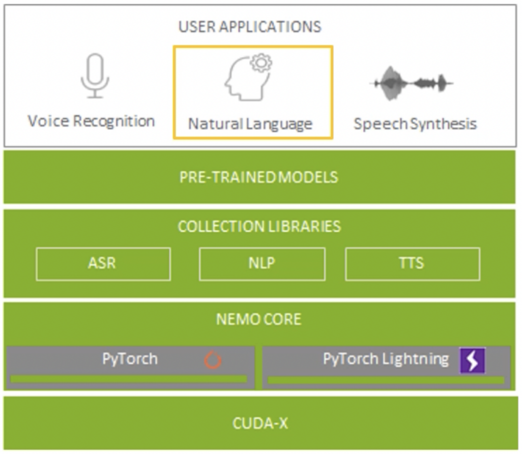


1.1 NeMo: A toolkit for Conversational AI

NeMo framework provides NLP capabilities along with Automatic Speech recognition (ARS), and Text-to-Speech (TTS).

NeMo

A Toolkit for Conversational AI



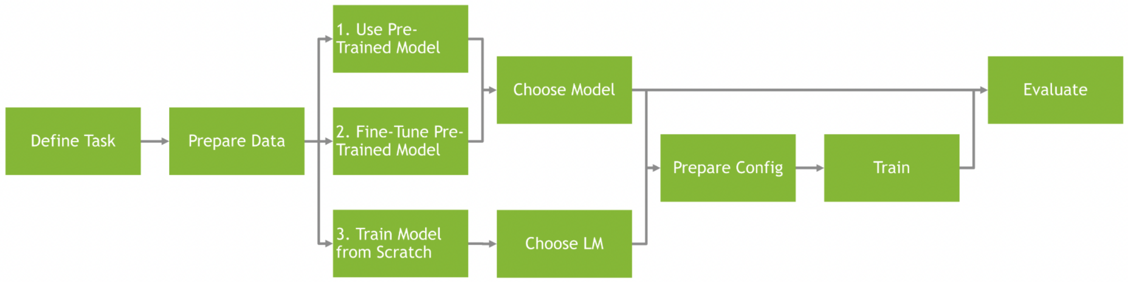
- Consists of collections for Automatic Speech Recognition (ASR), Natural Language Processing (NLP), and Text-to-Speech (TTS)
- Enables high-level APIs for training custom models to simplify the development with prebuilt modules
- Supports large catalogue of pre-trained models
- Uses PyTorch Lighting for easy and performant multi-GPU/multi-node mixed-precision training

Here is the official documentation for Nvidia's NeMo framework.

Using this pre-built component for recognition of token sections, we can perform tokenization, position embedding, padding, attention masking, and more. This allows us to build custom tokenization models.

1.2 Lab 1: Name-Entity Recognition Practice

Project Overview



1.2.1 Lab Overview and Steps

1. Dataset choice
2. NeMo Architecture Requirements
3. Using a Pre-trained model

Dataset Choice

The lab utilizes the Groningen Meaning Bank (GMB) which represents a large set of labelled classes including Geographical entity (LOC), Organization (ORG), Person (PER), Geopolitical Entity (GPE), Time indicator (TIME), Artifact (ART), Event (EVE), and Natural Phenomenon (NAT). Note: The dataset is not 100% human-labelled so the dataset is not considered entirely ground-truth. The data is also labelled using IOB format where each annotation has a prefix of **I**, **O**, and **B**.

NeMo Architecture Requirements

In order to feed in our ”ground-truth” labels for our pre-built module like BERT to learn the specific labels for each category that we would like to recognize, we need to feed in a `text.txt` and a `labels.txt` file for our entities and their corresponding labels.

This step assumes that the preprocessing of our label data is complete. Suppose that you have raw data, NeMo is compatible with Datasaur. Here is a tutorial for it. The NeMo model also includes built-in methods that take advantage of these capabilities: `TokenClassificationModel.add_predictions()` and `TokenClassificationModel.evaluate_from_file()` which are inherited from the `NLPModel`