IMDB Review Sentiment Analysis

Overview

This project aims to classify IMDB movie reviews as either positive or negative using various machine learning techniques. The process includes text preprocessing, generating text embeddings, and training supervised, semi-supervised, and self-supervised models. The main goal is to compare the performance of these models and determine the minimum amount of labeled data needed to achieve satisfactory results.

Libraries and Tools

NLTK (Natural Language Toolkit)

A library for natural language processing (NLP). In this project, it's used for:

Tokenizing text into words.

Removing stopwords (common words like "the", "and", etc., that do not carry significant meaning).

Lemmatizing words (converting words to their base or dictionary form).

Downloading necessary linguistic resources.

BeautifulSoup

A library for parsing HTML and XML documents. Here, it's used to remove HTML tags from the text data.

Scikit-learn

A machine learning library in Python. It's used for:

Logistic regression model: A simple yet effective linear model for binary classification.

Label propagation: A semi-supervised learning algorithm.

Calculating accuracy scores for model evaluation.

TensorFlow and TensorFlow Hub

TensorFlow is a deep learning library. TensorFlow Hub is a repository of pre-trained models. In this project, the Universal Sentence Encoder from TensorFlow Hub is used to generate text embeddings.

Transformers

A library for NLP models, including pre-trained models from the Hugging Face hub. While not directly used in the primary flow, it provides advanced tools for handling text data.

Zipfile

Used for handling compressed files, which is particularly useful for managing NLTK data files in environments like Kaggle.

Key Concepts Explained

1. Text Preprocessing

Text preprocessing is crucial for converting raw text into a format suitable for machine learning models. This includes several steps:

HTML Tag Removal: Using BeautifulSoup, HTML tags are stripped from the text to focus solely on the textual content.

Lowercasing: Text is converted to lowercase to ensure uniformity in word representations.

Punctuation Removal: All punctuation marks are removed to simplify the text and reduce noise.

Tokenization: The text is split into individual words (tokens) for further processing.

Stopword Removal: Common words (e.g., "the", "and") that do not contribute to sentiment analysis are removed.

Lemmatization: Words are reduced to their base or dictionary forms (e.g., "running" to "run"), reducing variations of words to their core meaning.

2. Self-Supervised Learning

Self-supervised learning is a form of unsupervised learning where the model generates its own supervision signals from the input data. In this project:

Universal Sentence Encoder: A pre-trained model from TensorFlow Hub is used to generate embeddings (numerical representations) of text. These embeddings capture semantic meaning, allowing the model to understand the context of sentences without explicit labels.

Propagated Labels: After generating embeddings for a small fraction of labeled data, these labels are propagated to the unlabeled data using techniques like label propagation. This pseudo-labeling allows the model to learn from both labeled and unlabeled data, enhancing its understanding of the sentiment in IMDB reviews.

3. Semi-Supervised Learning

Semi-supervised learning combines labeled and unlabeled data to improve model performance:

Label Propagation: Using techniques like LabelPropagation from scikit-learn, labels from a small fraction of labeled data are spread (propagated) to unlabeled data points based on their similarity in the embedding space. This expands the training set effectively without the need for extensive manual labeling.

Logistic Regression Model: A simple yet effective linear model is trained using the combined dataset (labeled and propagated labels). This model learns from both supervised labels and inferred labels, potentially improving accuracy compared to using only a small fraction of labeled data.