Multimodal Speech-text Satire Recognition in Spanish

Nguyen Minh Bao

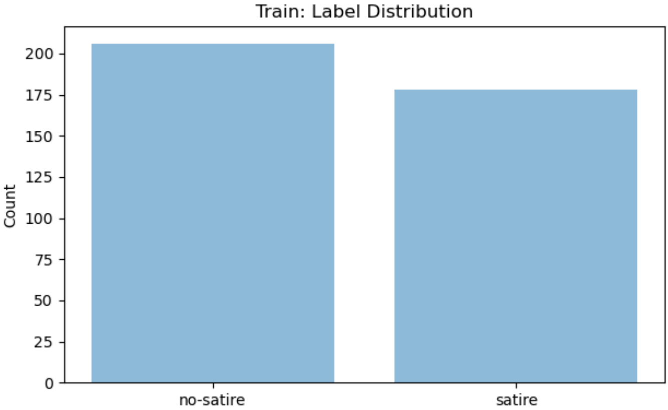
April 2025

Introduction

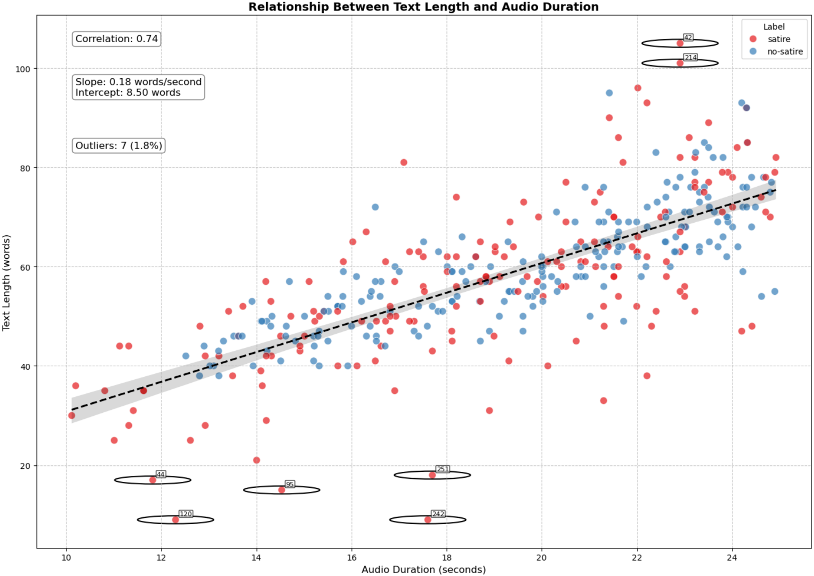
- **Problem**: Classify satire or non-satire in Spanish using both speech and text.
- **Missions**:
  - **Task1:** Text Satire Detection(only Text)
  - **Task2:** Multimodal Satire Detection(Audio + Text)
  - **Task3:** Audio Satire Detection
- Dataset:
  - training set: 386 samples
  - validation set: 96 samples
  - testing set: 6000 samples

Exploratory Data Analysis (EDA)

- Total samples: 386 points



Exploratory Data Analysis (EDA)



Preprocessing Data

1. **Label Encoding:**
    - Converts categorical labels into numerical values for machine learning models.
    - Example: - `satire` → `1` - `non-satire` → `0`
    - Makes it easier for models to understand and classify the data.
2. **Word tokenize:**
    - Splits text data into individual words or tokens, which are essential for text analysis and feature extraction in natural language processing (NLP).
    - Example: - Sentence: `"This is a satire article."` - Tokens: `["This", "is", "a", "satire", "article", "."]`

Feature Extraction for Text

- Bag of words
- TF-IDF
- Word2vec

Bag of Words

1. **max_features:**
   - Limits the vocabulary size to the most frequent words.
   - Example: `max_features=5000` keeps only the top 5000 most common words.
   - Reduces dimensionality and computational cost.
2. **ngram_range:**
   - Defines the range of n-grams to include (e.g., single words or phrases).
   - Example: `ngram_range=(1,2)` includes both unigrams (1-grams) and bigrams (2-grams).
   - Captures context and relationships between words.
3. **lowercase:**
   - Converts all text to lowercase for consistency.
   - Prevents treating words like "The" and "the" as different tokens.

TF-IDF: Term Frequency-Inverse Document Frequency

Definition: TF-IDF measures the importance of a term in a document relative to a collection of documents (corpus).

Max Features:

- The max_features parameter limits the number of features (words) considered by selecting the top most important terms based on their TF-IDF scores.
- Max_features=5000, only the 5000 most relevant terms will be included in the feature matrix, reducing dimensionality and computational cost.

Definition: Word2Vec is used to encode text into dense vector representations (word embeddings) using the Gensim library.

Key Parameters:

- **vector_size=100:**
- **window=10:**
- **min_count=1:**

Librosa - Audio Feature Extraction

Key Features Extracted by Librosa:

- Spectral Features:
  - Spectral Centroid
  - Spectral Bandwidth
  - Spectral Rolloff
- Time-Domain Features:
  - Zero-Crossing Rate
  - RMS Energy
- Mel-Frequency-Based Features:
  - MFCCs (Mel-Frequency Cepstral Coefficients)
  - Mel Spectrogram
- Chroma Features:
  - Chroma STFT

Combine text and audio: using the function **concatenate** to merger vector of text and audio

Training Models -Task1 - Using Bag of Words

Training models:

Model	Accuracy	Training Parameters
SVM (Linear Kernel)	92.71%	kernel=linear, C=0.001
SVM (Poly Kernel)	91.67%	kernel=poly, C=10, coef0=1, degree=2
SVM (RBF Kernel)	89.58%	kernel=rbf, C=20, gamma=10^{-5}
Logistic Regression	94.79%	solver=lbfgs, C=0.01
Naive Bayes	96.88%	alpha=0.5, $\text{fit}_p rior = False$

Evaluate models:

Model	Accuracy	Training Parameters
SVM (Linear Kernel)	83.02%	kernel=linear, C=0.001
SVM (Poly Kernel)	83.25%	kernel=poly, C=10, coef0=1, degree=2
SVM (RBF Kernel)	81.95%	kernel=rbf, C=20, gamma=10^{-5}
Logistic Regression	84.20%	solver=lbfgs, C=0.01
Naive Bayes	84.47%	alpha=0.5, $\text{fit}_p rior = False$

Training Models -Task1 - TF-IDF

Training models:

Model	Accuracy	Training Parameters
SVM (Linear Kernel)	93.75%	kernel=linear, C=0.001
SVM (Poly Kernel)	94.79%	kernel=poly, C=10, coef0=0.1, degree=2
SVM (RBF Kernel)	93.75%	kernel=rbf, C=100, gamma=10^{-6}
Logistic Regression	94.79%	solver=lbfgs, C=0.01
Naive Bayes	95.83%	alpha=0.1, fitprior=False

Evaluate models:

Model	Accuracy	Training Parameters
SVM (Linear Kernel)	83.72%	kernel=linear, C=0.001
SVM (Poly Kernel)	84.15%	kernel=poly, C=10, coef0=0.1, degree=2
SVM (RBF Kernel)	84.10%	kernel=rbf, C=100, gamma=10^{-6}
Logistic Regression	84.32%	solver=lbfgs, C=0.01
Naive Bayes	84.50%	alpha=0.1, fitprior=False

Training Models -Task1 - Word2vec

Training models:

Model	Accuracy	Training Parameters
SVM (Linear Kernel)	87.50%	kernel=linear, C=1
SVM (Poly Kernel)	83.33%	kernel=poly, C=10, coef0=1.0, degree=2
SVM (RBF Kernel)	80.21%	kernel=rbf, C=30, gamma="scale"
Logistic Regression	87.50%	solver=lbfgs, C=1
Naive Bayes	68.75%	alpha=0.001, fitprior=False

Evaluate models:

Model	Accuracy	Training Parameters
SVM (Linear Kernel)	79.03%	kernel=linear, C=1
SVM (Poly Kernel)	79.20%	kernel=poly, C=10, coef0=1.0, degree=2
SVM (RBF Kernel)	79.13%	kernel=rbf, C=30, gamma="scale"
Logistic Regression	79.97%	solver=lbfgs, C=1
Naive Bayes	66.02%	alpha=0.001, fitprior=False

Training Models -Task2 - BoW with Librosa - MFCCs

Training models:

Model	**Accuracy**	**Training Parameters**
SVM (Linear Kernel)	95.83%	kernel=linear, C=0.001
SVM (Poly Kernel)	92.71%	kernel=poly, C=1, coef0=1.0, degree=3
SVM (RBF Kernel)	89.58%	kernel=rbf, C=20, gamma=10^{-5}
Logistic Regression	94.79%	solver=lbfgs, C=0.01
Naive Bayes	84.38%	alpha=1.0, fitprior=False

Training Models -Task2 - TF-IDF with Librosa - MFCCs

Training models:

Model	Accuracy	Training Parameters
SVM (Linear Kernel)	95.83%	kernel=linear, C=0.001
SVM (Poly Kernel)	95.83%	kernel=poly, C=0.1, coef0=1.0, degree=4
SVM (RBF Kernel)	94.79%	kernel=rbf, C=5, gamma=10^{-4}
Logistic Regression	95.83%	solver=lbfgs, C=0.01
Naive Bayes	92.71%	alpha=1.0, fitprior=False

Training Models -Task2 - Word2vec with Librosa - MFCCs

Training models:

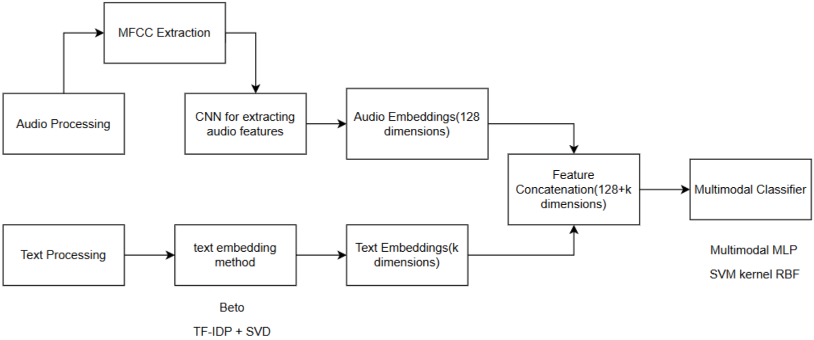
Model	Accuracy	Training Parameters
SVM (Linear Kernel)	93.75%	kernel=linear, C=0.1
SVM (Poly Kernel)	90.62%	kernel=poly, C=1, coef0=1.0, degree=2
SVM (RBF Kernel)	91.67%	kernel=rbf, C=30, gamma=10^{-3}
Logistic Regression	93.75%	solver=lbfgs, C=1
Naive Bayes	78.12%	alpha=1.0, fitprior=False

Training Models -Task3 - Librosa

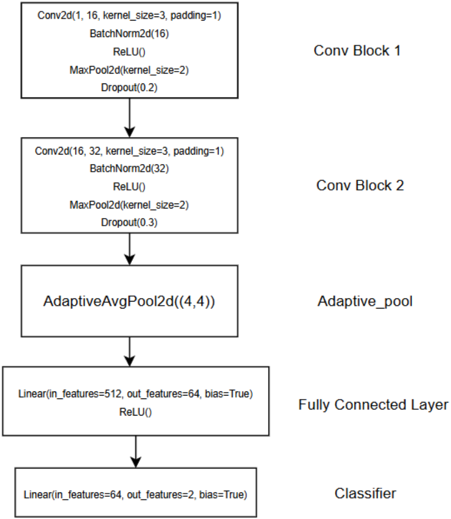
Training models:

Model	Accuracy	Training Parameters
SVM (Linear Kernel)	89.58%	kernel=linear, C=0.1
SVM (Poly Kernel)	88.54%	kernel=poly, C=100, coef0=0.1, degree=3
SVM (RBF Kernel)	92.71%	kernel=rbf, C=5, gamma=10^{-2}
Logistic Regression	91.67%	solver=lbfgs, C=1
Naive Bayes	71.88%	alpha=0.01, fitprior=False

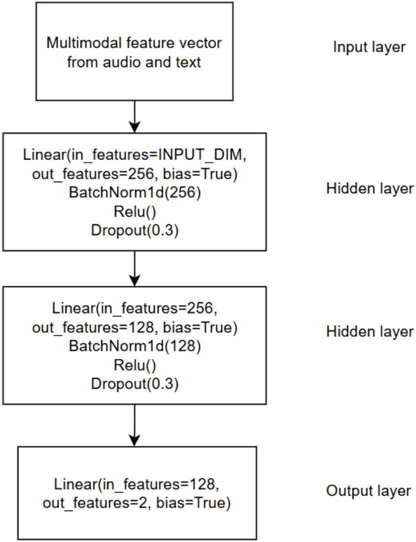
Pipeline using CNN to get features of audio in Task2



The structure of CNN



The structure of MultimodalMLP



Results of pipeline

TF-IDP + SVD

Model	**Accuracy**	**Training Parameters**
Multimodal MLP	80.3845%	
SVM (RBF Kernel)	81.1747%	kernel=rbf, C=10, gamma=10^{-3}

BETO

Model	**Accuracy**	**Training Parameters**
Multimodal MLP	83.5854%	
SVM (RBF Kernel)	82.9076%	kernel=rbf, C=5, gamma=10^{-3}

Summary of training models

Task1:

- BoW and TF-IDF with Naive bayes have the best performance
- Word2Vec: SVM and Logistic regression have stable accurary on testing datas.

Task2:

- The performance of all models are improved powered by the audio feature(MFCCs)
- If we add more audio feature to training the model will have the high variance so the performance of models can be reduced

Task3:

- The models using only audio features with the previous 8 features achieved the best performance.

Q&A

Thank you for your attention!