Predicción de tópicos a partir de un archivo de audio

Topic prediction from an audio file

Autor 1: David Cediel Gómez Autor 2: Santiago Londoño, Autor 3: Juan Pablo Narváez
Facultad de Ingenierías, Universidad tecnológica de Pereira, Pereira, Colombia
Correo-e: david.cediel@utp.edu.co,

***Resumen*— Uno de los principales temas del Machine Learning y de la Inteligencia Artificial es el procesamiento del lenguaje Natural (NLP por sus siglas en inglés), dentro de este tema se encuentra la clasificación de textos, la cual etiqueta textos con un tópico específico según su contenido, en este artículo se ilustrará la manera de realizar la clasificación de audios noticias a partir de 5 tópicos.**

***Palabras clave*— *machine learning, tópico, procesamiento, preprocesamiento, lenguaje natural, lema, palabras inútiles, limpieza, unigrama, importancia, contexto, tokens, texto*.**

***Abstract*— One of the main topics of Machine Learning and Artificial Intelligence is Natural Language Processing (NLP), within this theme is the classification of texts, which labels texts with a specific topic according to their content This article will illustrate how to perform the classification of audio news from 5 topics.**

***Key Word* — machine learning, topic, processing, preprocessing, natural language, motto, useless words, cleanliness, unigram, importance, context, tokens, text.at** http://www.ieee.org/web/developers/webthes/index.htm**.**

I. INTRODUCCIÓN

El procesamiento del lenguaje natural es utilizado en diversos ámbitos: análisis de discursos, resumen de textos, análisis de sentimientos, etc. Dentro de este tema esta la clasificación de textos, este último es el enfoque de este artículo.

En el proyecto se eligieron 5 tópicos en los cuales clasificar los textos, estos fueron: deportes, entretenimiento, negocios (economía), política y tecnología. Se describirá la forma en que se realizó y que datos se usaron.

En el artículo se tratarán varios temas:

1. Se dará una breve introducción de qué es el lenguaje natural, cuales son sus principales elementos y cómo se realiza una limpieza del texto antes de su análisis (Eliminación de stop-words y yendo a la raíz de la palabra con la lematización)
2. Se mencionará cuáles fueron los datos usados y cómo se realizó la limpieza y la extracción de características principales.
3. Se mencionará cómo se entrenó el modelo y cuáles fueron los resultados de este entrenamiento
4. Se mencionará cómo se realizó la conexión del back-end con la funcionalidad para entrar desde un navegador web

II. CONTENIDO

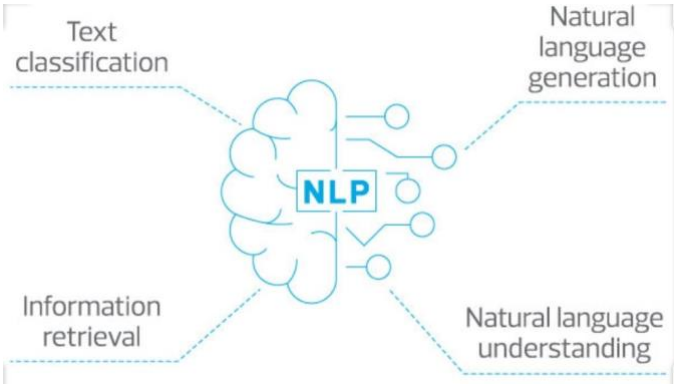
A. PROCESAMIENTO DEL LENGUAJE NATURAL

Es definido como un campo de las ciencias de la computación, más específicamente de la inteligencia artificial que estudia las interacciones entre las computadoras y los lenguajes naturales, como el español, inglés o el chino. Este se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la interpretación del lenguaje. En un principio los sistemas de PLN se basaban en un complejo conjunto de reglas diseñadas a mano (modelos lógicos o gramáticas). Pero a partir de la década de 1980 se dio una revolución en este campo con el uso de algoritmos de aprendizaje automático (modelos probabilísticos).

Componentes del lenguaje natural:

1. Análisis morfológico o léxico: Esta parte del proceso consiste en el análisis interno de las palabras que forman oraciones para extraer lemas, rasgos flexivos, unidades léxicos. Es esencial para la formación básica: Categoría sintáctica y significado léxico.
2. Análisis sintáctico: Consiste en el análisis de la estructura de las oraciones de acuerdo con el modelo gramatical empleado (lógico o estadístico).
3. Análisis semántico: Proporciona la interpretación de las oraciones, una vez eliminadas las ambigüedades morfosintácticas
4. Análisis pragmático: Incorpora el análisis del contexto de uso a la interpretación final. Aquí se incluye el tratamiento del lenguaje figurado (metáfora e ironía) como el conocimiento del mundo especifico necesario para entender un texto especializado.

Fecha de Recepción: 13-Nov-2019
Fecha de Aceptación:



$$TFIDF(t,d) = TF(t,d) \times \log\left(\frac{N}{DF(t)}\right)$$

Being:

- t: term (i.e. a word in a document)
- d: document
- $TF(t)$: term frequency (i.e. how many times the term t appears in the document d)
- N: number of documents in the corpus
- $DF(t)$: number of documents in the corpus containing the term t

Imagen 1. Explicación de TFIDF

A partir de este, se puede convertir el texto en datos numéricos, mostrando una representación de las palabras más frecuentes e importantes de cada documento, esta visualización se realizó a través de la librería chi2 que extrae características, los resultados obtenidos fueron los siguientes.

B. DATA SET Y PREPROCESAMIENTO

El data set usado es una recopilación de noticias de la cadena de televisión de la BBC de Inglaterra, este se encontraba dividido por carpetas en las cuales se encontraba alrededor de 400 archivos de texto plano de cada tópico.

Para poder realizar el modelo, se debieron traducir todos los archivos de texto a español, ya que estaban en inglés, para realizar esto, se realizó un script de Python que, con la ayuda de la librería de la librería googletrans se tradujo todo el dataset, el código puede ser consultado en el archivo "pruebaTraduccion.py".

Luego de traducir los archivos de texto, se etiquetaron con su respectivo tópico y guardaron en formato texto → etiqueta en un archivo tipo pickle, el código puede ser consultado en el archivo "crearDataset.py"

A partir del conjunto de datos organizados y etiquetados, se realizó un preprocesamiento, en el cual se quitaron las tildes de las palabras, se eliminaron espacios repetidos, se quitaron todos los símbolos que podría tener el texto, además, se cambió la palabra por su lema y se quitaron las palabras inútiles para tener un modelo sin tanto ruido.

Por último, antes de ingresar los datos al modelo, se usó un método de extracción de características principales llamado **tfidf** (Term frequency – Inverse document frequency) el cual extrae los términos más frecuentes del documento

```
# 'business' category:
  . Palabras mas correlacionadas:
. economia
. banco
. accionar
. crecimiento
. bn
  . Dos palabras mas correlacionadas:
. ano edad
. mil millón

# 'entertainment' category:
  . Palabras mas correlacionadas:
. cine
. estrellar
. premio
. actor
. pelicula
  . Dos palabras mas correlacionadas:
. reino unido
. mil millón
```

Imagen 2. Términos más importantes de los tópicos

```
# 'politic' category:
  . Palabras mas correlacionadas:
. elección
. liberal
. ministro
. blair
. conservador
  . Dos palabras mas correlacionadas:
. ano edad
. reino unido

# 'sport' category:
  . Palabras mas correlacionadas:
. temporada
. equipar
. jugar
. jugador
. victoria
  . Dos palabras mas correlacionadas:
. reino unido
. ano edad
```

Imagen 3. Términos más importantes de los tópicos

```
# 'tech' category:
  . Palabras mas correlacionadas:
. red
. ordenador
. software
. usuario
. tecnologia
  . Dos palabras mas correlacionadas:
. ano pasar
. ano edad
```

Imagen 4. Términos más importantes de los tópicos

Con este método, se convierte el texto en un arreglo numérico que puede ser enviado como entrada para el modelo

C. ENTRENAMIENTO DEL MODELO Y RESULTADOS

El modelo realizado, se construyo una cuadrícula con hiperparámetros para construir una máquina de vector de soporte para la clasificación, en total se probaron 84 modelos distintos, el mejor modelo es mostrado a continuación.

```
Mejores parametros para el modelo luego de la busqueda:
{'C': 0.1, 'kernel': 'linear', 'probability': True}

Acc promedio del mejor modelo:
0.9406099518459069
```

Imagen 5. Hiperparámetros del mejor modelo y su desempeño

Como se puede observar en la anterior imagen, el desempeño promedio del modelo en la etapa de entrenamiento dio 94.06% y el desempeño con los datos de prueba fue de 94.01%.

Por último, se guarda el modelo con la ayuda de la librería joblib, ya con el modelo entrenado, solo se debe cargar y predecir para las operaciones.

D. USO DEL FRAMEWORK WEB

Para el manejo del framework se utilizo Django que es un framework de código abierto, escrito en Python, que hace uso del patrón de diseño conocido como modelo-vista-template. Su uso se da porque su manejo facilita y agiliza la creación de sitios web.
Su fácil manejo se puede evidenciar en el archivo audio.py donde se maneja una pagina como una clase principal con el nombre de SubirAudio, donde se tienen como funciones get_context_data que recibe como argumentos ** kwargs (que permite pasar argumentos de longitud variable de palabras clave a una función), e inicializa la pagina. Get_success_url que permite ver el audio si la conexión es exitosa. Y Post que se encarga de recibir los audios, guardarlos, hacer su conversión a texto por medio de la librería volver texto y teniendo el texto del audio respectivo predice el tema por medio de analizar que hace parte de la librería. .predictorBackEnd y por último guarda el tema y permite su visualización.

III. CONCLUSIONES

Como primera conclusión tenemos que para este tipo de trabajos el ruido es un factor muy importante a tener en cuenta ya que es muy fácil que los programas que se usan para pasar de audio a texto sean afectados por este.
Como se menciona anteriormente el uso de Django facilita en gran medida la programación de páginas web y es un framework recomendable para esta práctica.
Al usar modelos probabilísticos para el procesamiento del lenguaje natural este no llega a ser 100% verídico pero si llega a presentar un alto porcentaje de fiabilidad.

REFERENCIAS

- https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales
- http://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/
- https://es.wikipedia.org/wiki/Django_(framework)
- https://www.djangoproject.com/start/overview/
- https://docs.djangoproject.com/en/2.2/topics/http/views/