

Análisis de voz y texto en Python para reconocimiento de emociones

Paula Andrea Pérez Toro

Ingeniera Electrónica

Estudiante de maestría en ingeniería

June 11, 2019

GITA research group, University of Antioquia.

paula.perez@udea.edu.co



Outline

Emociones

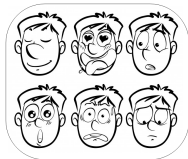
Análisis de voz en el reconocimiento de emociones

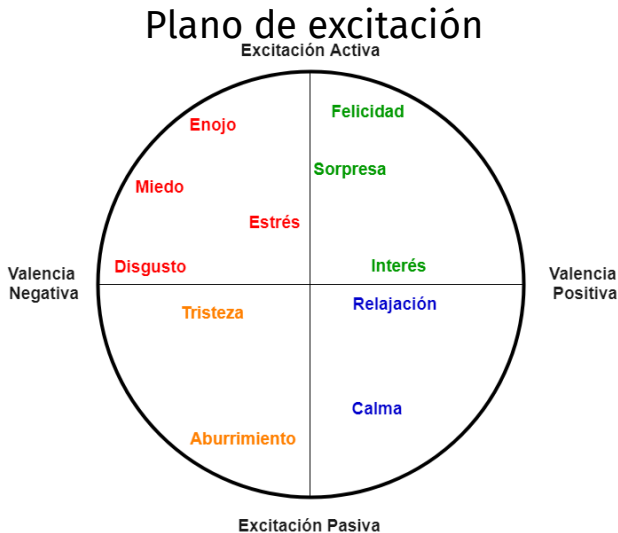
Análisis de texto en el reconocimiento de emociones

Emociones

Emociones

- Los seres humanos pueden expresar cerca de 27 emociones.
- Es un método útil para discriminar procesos cognitivos relacionados con la arquitectura de la mente humana.





Modelamiento de emociones

Aspectos “Paralingüísticos”



Información no verbal

Lenguaje Natural



Información Verbal

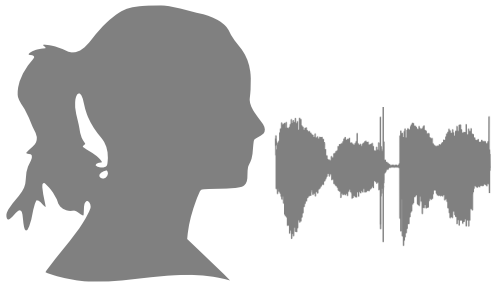
Análisis de voz en el reconocimiento de emociones

Señal de voz



Tengo hambre, quiero una pizza.

Señal de voz



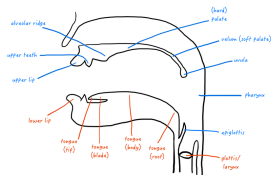
- Respiración
- Fonación
- Resonancia
- **Articulación**
- Prosodia

Tengo hambre, quiero una pizza.

Cambios psicológicos están influenciados por aspectos emocionales afectan la producción de la voz .

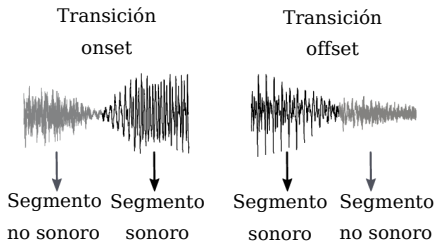
Caracterización de la voz clásica: Articulación

Articulación: proceso por el cual el sonido, generado en las cuerdas vocales, se modifica por efecto de los movimientos de los órganos articuladores.



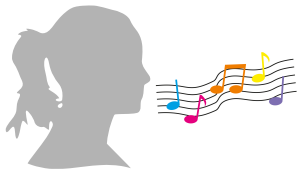
Caracterización:

- Transiciones onset y offset
- Formantes
- Energía de la bandas de Bark
- Coeficientes Ceptrales de Mel



Caracterización de la voz clásica: Prosodia

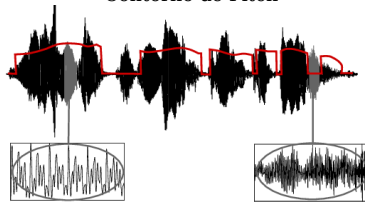
Prosodia: fenómenos fónicos que abarcan más de un fonema o segmento -entonación, acentuación, ritmo, velocidad de habla, entre otras



Caracterización:

- Transiciones onset y offset
- Energía
- Pitch (Tono)
- Duración
- Velocidad

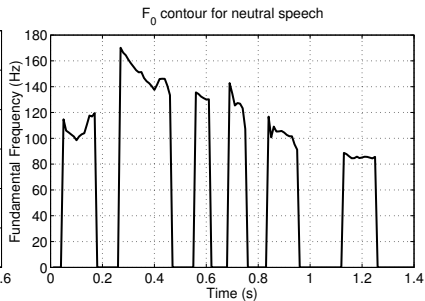
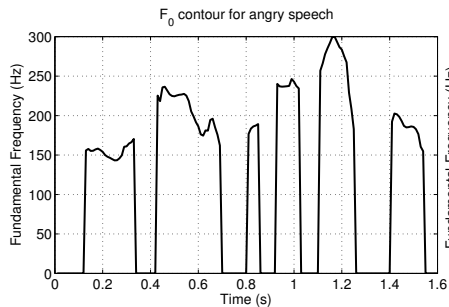
Contorno de Pitch



Fragmento sonoro

Fragmento no sonoro

Caracterización de la voz clásica: Prosodia



Disvoice

Disvoice es una librería en python opensource que se encuentra en github, para el análisis de las señales de voz. Esta contiene tres grandes grupos de características:

- Fonación
- Articulación
- Prosodia

Desarrollado por: Juan Camilo Vasquez Correa - Grupo GITA

Link github: <https://github.com/jcvasquezc/DisVoice>

Disvoice: Articulación

```
python articulation.py <file_or_folder_audio> <file_features.txt> [dynamic_or_static (default static)] [plots (true or false) (default false)] [kaldi output (true or false) (default false)]
```



```
python articulation.py "/home/camilo/Camilo/data/BDKayElementrics/Norm/Rainbow/" "featuresDDKstatFolder.txt" "dynamic" "false" "true"
```

Disvoice: Prosodia

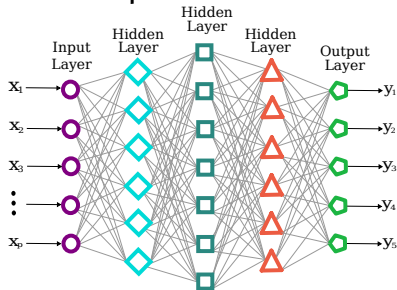
```
python prosody.py <file_or_folder_audio> <file_features.txt> [dynamic_or_static (default static)] [plots (true or false) (default false)] [kaldi output (true or false) (default false)]
```



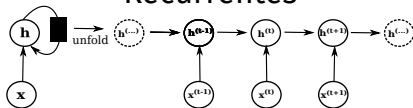
```
python prosody.py "/home/camilo/Camilo/data/BDKayElemetrics/Norm/Rainbow/" "featuresDDKstatFolder.txt" "dynamic" "false" "true"
```

Métodos basados en redes neuronales para el análisis de la voz

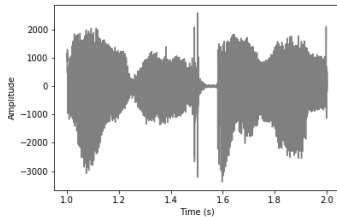
Redes neuronales profundas



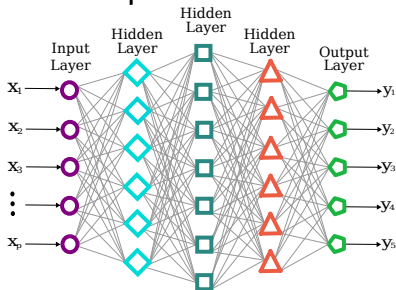
Recurrentes



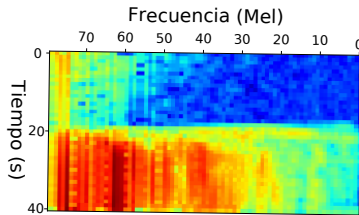
Serie de tiempo



Redes neuronales profundas



Espectro de la voz



Convolucionales



Algunas librerías de Python

- Scipy
- Python speech features
- openSmile (bash)
- Librosa
- Disvoice
- Pytorch
- TensorFlow
- Keras



PYTORCH



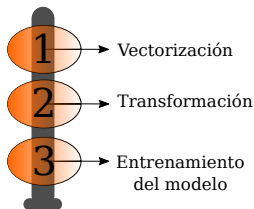
Análisis de texto en el reconocimiento de emociones

Procesamiento de Lenguaje Natural



Análisis de texto en el reconocimiento de emociones

NLP consta de procesos sistemáticos para analizar y entender el lenguaje humano derivado de textos.



Esquema de tubería para NLP ([pedregosa2011scikit](#))

Pre-procesamiento

Eliminando signos de puntuación y stop words:

```
1 #Creando un oración de test
2 test_sentence='Wow!! Feliz escribiendo mi primer texto para NLP!!'
3 #Eliminando los signos de puntuación
4 nopunctuation= [char for char in test_sentence
5                  if char not in string.punctuation]
6 nopunctuation
7 #Now eliminate the punctuation and print them as a whole sentence
8 nopunctuation=''.join(nopunctuation)
9 nopunctuation
10
11 'Wow Feliz escribiendo mi primer texto para NLP'
```

Pre-procesamiento

Eliminando signos de puntuación y stop words:

```
1 #Dividimos cada palabra de la oración resultante
2 nopunctuation.split()
3 #Eliminamos las StopWords
4 clean_sentence= [word for word in nopunctuation.split() if
5 word.lower() not in stopwords.words('spanish')]
6 clean_sentence
7
8 ['Wow', 'Feliz', 'escribiendo', 'primer', 'texto', 'NLP']
```

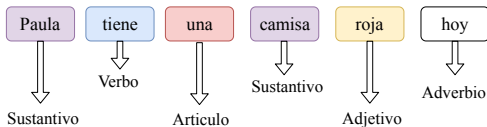
Pre-procesamiento

Lematización:

```
1  import spacy
2  nlp = spacy.load('es_core_news_sm')
3  #clean_sentence=''.join(clean_sentence)
4  doc = nlp(clean_sentence[2])
5
6  tokenLemma=[]
7  for token in doc:
8      print(token, '-->', token.lemma_)
9      tokenLemma.append(token.lemma_)
10
11  escribiendo --> escribir
```

Métodos clásicos del análisis de texto

- Análisis sintáctico
(Syntactic parsing)
- Conteo de palabras y características estadísticas
- Métodos basados en reglas



Uno de los métodos más populares es **Part of Speech Tagging (PoS)**

Métodos clásicos del análisis de texto

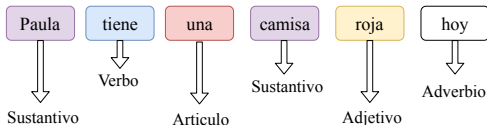
- Análisis sintáctico (Syntactic parsing)
- Conteo de palabras y características estadísticas
- Métodos basados en reglas

Métodos más populares como:
Bag of Words (BoW),
Term Frequency-
Inverse Document
Frequency (TF-IDF)

"La mayoría de las gaviotas no se molesta en aprender sino las normas de **vuelo** más elementales: como ir y volver entre playa y comida. Para la mayoría de las gaviotas, no es **volar** lo que importa, sino comer. Para esta gaviota, sin embargo, no era comer lo que le importaba, sino **volar**. Más que nada en el mundo, Juan Salvador Gaviota amaba **volar**."

Métodos clásicos del análisis de texto

- Análisis sintáctico
(Syntactic parsing)
- Conteo de palabras y
características
estadísticas
- Métodos basados
en reglas



Ejemplos Métodos clásicos NLP

Part of Speech Tagging:

```
1 #Part of Speech Tagging
2 from spacy.lang.es.examples import sentences
3 import spacy
4
5 nlp = spacy.load('es_core_news_sm')
6 doc = nlp('Yo estoy comiendo pizza con Paula')
7 print(doc.text)
8 for token in doc:
9     print(token.text, token.pos_, token.dep_)
10
11 Yo estoy comiendo pizza con Paula
12 Yo PRON nsubj
13 estoy AUX aux
14 comiendo VERB ROOT
15 pizza VERB obj
16 con ADP case
17 Paula PROPN obl
```

Ejemplos Métodos clásicos NLP

Bag of Words:

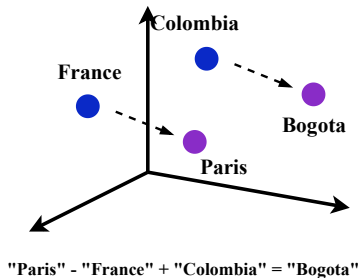
```
1  #Librerias
2  from sklearn.feature_extraction.text import CountVectorizer
3  #to tokenize the document
4  #Vectorización
5  vectorizer= CountVectorizer()
6  #Creación de documentos
7  document1='Hola como estas'
8  document2='hoy es un muy muy muy buen dia y nosotros podemos
9  tener diversion diversion diversion'
10 document3= 'Esta fue una experiencia emocionante'
11 listofdocuments=[document1, document2, document3]
```

Ejemplos Métodos clásicos NLP

Bag of Words:

```
1  #Entrenar BoW
2  bag_of_words=vectorizer.fit(listofdocuments)
3  # TransformacionBoW
4  bag_of_words=vectorizer.transform(listofdocuments).toarray()
5  bag_of_words2=vectorizer.transform(listofdocuments)
6  #Imprimir BoW
7  print(bag_of_words)
8  print(bag_of_words2)
9
10 [[0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0]
11    [1 0 1 3 0 1 0 0 0 0 0 1 3 1 1 1 1 0]
12    [0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 1]]
```

Word Embeddings



- Word2Vec
- Glove
- ELMo
- BERT

One-Hot encoding:
Colombia=[1 0 0 0]

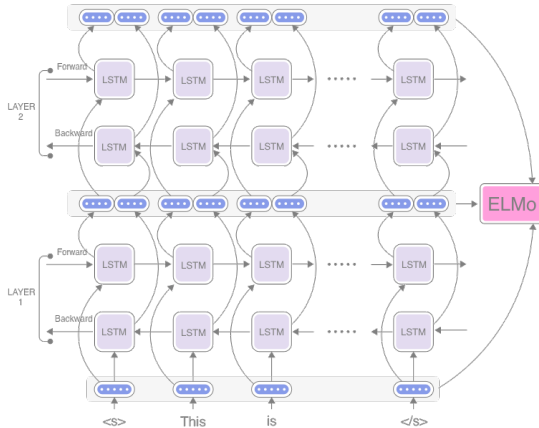
Word Embeddings

Considerando una ventana de uno: “Yo amo Programar. Yo amo Matemáticas. Yo tolero Biología.”

	Yo	amo	Programar	Matematicas	tolero	Biologia	.
Yo	0	2	0	0	1	0	2
amo	2	0	1	1	0	0	0
Programar	0	1	0	0	0	0	1
Matematicas	0	1	0	0	0	0	1
tolero	1	0	0	0	0	1	0
Biologia	0	0	0	0	1	0	1
.	2	0	1	1	0	1	0

- Word2Vec
- Glove
- ELMo
- BERT

Word Embeddings



- Word2Vec
- Glove
- **ELMo**
- **BERT**

Algunas librerías de Python

- Spacy
- NLTK: Natural Language Toolkit
- SckitLearn
- Gensim
- Glove
- FastText
- Pytorch, Keras, TensorFlow



Gracias por su atención

contacto: paulaperez16@gmail.com