# Clasificación de Subjetividad utilizando Técnicas de Aprendizaje Automático

Juan Manuel Coria

# Estructura de la Presentación

1. Problemática
2. Definiciones
3. Estudio de técnicas y algoritmos
4. Estado del arte
5. Construcción de una base de datos
6. Diseño de una solución
7. Conclusiones

# Problemática

Hoy existen muchos estudios, técnicas, herramientas y bases de datos que atacan el problema del análisis de subjetividad en inglés.

Existen muy pocos (ninguno) que tomen una perspectiva completa desde el castellano.

El objetivo de este trabajo fue el de crear un modelo que contemple el español de principio a fin, junto con una base de datos y una implementación que lo soporten.

# Definiciones

# Subjetividad y Objetividad

Tomando en cuenta diferentes definiciones de estos conceptos: la RAE, Wiebe y Liu, se decidió considerar las siguientes definiciones:

- Una oración subjetiva expresa algún tipo de pensamiento, creencia, sentimiento o percepción relativa al sujeto. (Wiebe)
- Una oración objetiva expresa información factual acerca del mundo. (Liu)

Las definiciones de la RAE son muy genéricas, no contemplan el estado interno del sujeto. Wiebe da una definición muy precisa de subjetividad, orientada a textos literarios, y Liu tiene una visión más analítica y genérica.

# Técnicas y Algoritmos

- Tokenización
- Stopwords
- Stemming y Lematización
- Etiquetado Gramatical
- TF-IDF

- Support Vector Machines
- Multiperceptrón

# Tokenización

División del documento en términos o tokens.

Nuestro idioma permite esta separación fácilmente, dado que no hay contracciones como en inglés o francés, y existen términos individuales que pueden reconocerse fácilmente tanto por una persona como por una máquina: las palabras. Este no es el caso de otros idiomas como el chino o el japonés donde las palabras como las conocemos simplemente no existen.

# Stopwords

Palabras del idioma que carecen de información importante.

Se utilizó la lista de stopwords proveída por la librería nltk en Python (Natural Language Toolkit)

Incluye las conjugaciones de ser, estar, haber, tener, así como pronombres, artículos, etc.

Lista Completa: 314 palabras

# Stemming y Lematización

El objetivo de estas técnicas es modificar términos sintácticamente diferentes y semánticamente equivalentes para poder analizar sencillamente el significado de un texto.

El stemming es simple, elimina terminaciones basándose en reglas establecidas.

La lematización es más compleja, realiza un análisis léxico y lingüístico.

Las técnicas de stemming son más utilizadas por su bajo costo computacional y similares resultados prácticos.

# Stemming y Lematización

Existen diferentes algoritmos de stemming en inglés: Lovins (1968), Porter (1980), Paice-Husk (1990).

Snowball es un lenguaje creado para la definición e implementación de stemmers. Existen stemmers de Snowball para diferentes idiomas, entre ellos uno para español, que es utilizado en el trabajo.

# Etiquetado Gramatical (POS Tagging)

Es una técnica que requiere de un entendimiento gramatical y lingüístico.

Consiste en identificar la función de cada palabra en el texto, por ejemplo, si es un Sustantivo, Adjetivo, Adverbio, Pronombre, etc.

Esto es muy útil para la detección de subjetividad porque los modificadores (adjetivos y adverbios) generalmente tienen una alta influencia en la misma.

Obviamente no hay que fiarse de esto porque muchas frases pueden hablar indirectamente: *Daniel dice que hoy hace demasiado calor*.

# TF-IDF

Es una métrica que mide la relevancia de un término en un corpus de documentos.

# TF-IDF

Está dada por:

$$TF(x) = f_x / n$$
$$IDF(x) = \log_e(N / F_x)$$
$$TFIDF(x) = TF(x) * IDF(x)$$

$f_x$ es la frecuencia absoluta del término x en el documento, n la cantidad de términos en el documento, N es la cantidad total de documentos y $F_x$ la cantidad de documentos donde aparece el término x.

# TF-IDF

Es una métrica interesante, pero no pareciera ser muy relevante para la detección de subjetividad, sobre todo porque en este trabajo, el concepto de documento es inexistente, la base de datos es una lista de oraciones con su correspondiente etiqueta de subjetividad.

Se define entonces el SWF-ISF: Subjectivity Word Frequency - Inverse Sentence Frequency, que intenta medir el impacto de un término en la subjetividad de una oración.

# SWF-ISF

Se define como:

$$SWF(x) = f_{xs} / n_s$$
$$ISF(x) = \log_e(n / f_x)$$
$$SWFISF(x) = SWF(x) * ISF(x)$$

$f_{xs}$ es la cantidad de oraciones subjetivas en las que aparece la palabra x, $n_s$ la cantidad total de oraciones subjetivas, $n$ es la cantidad total de oraciones en la base de datos, y $f_x$ la cantidad de oraciones donde aparece la palabra x.

# SWF-ISF

Ejemplo: Asumiendo una base de datos con 25 oraciones subjetivas y 25 objetivas, donde las siguientes 2 oraciones son las únicas donde aparece la palabra largo.

1. Su cabello era largo y hermoso (Subjetiva)
2. El Nilo es un río largo que se extiende desde el lago Victoria hasta el Mar Mediterráneo (Objetiva)

$$\text{SWF("largo")} = 1 / 25 = 0.04$$
$$\text{ISF("largo")} = \log_e(50 / 2) = 3.219$$
$$\text{SWFISF("largo")} = 0.04 * 3.219 = 0.129$$

Como es de esperarse, la palabra largo no tiene un fuerte impacto sobre la subjetividad.

# SWF-ISF

Es importante notar que no se utiliza la longitud de la palabra para ponderar el SWF, sino la cantidad de oraciones subjetivas donde esta aparece.

Esto es porque las oraciones son de longitud variable, y no es deseable ponderar la presencia subjetiva de la palabra en base a la longitud de la oración. El valor de la métrica incrementaría considerablemente en oraciones más cortas.

Midiendo el SWF de esta forma, el valor final del SWF-ISF de *largo* en el ejemplo anterior sería 0.538 (en comparación a 0.129).

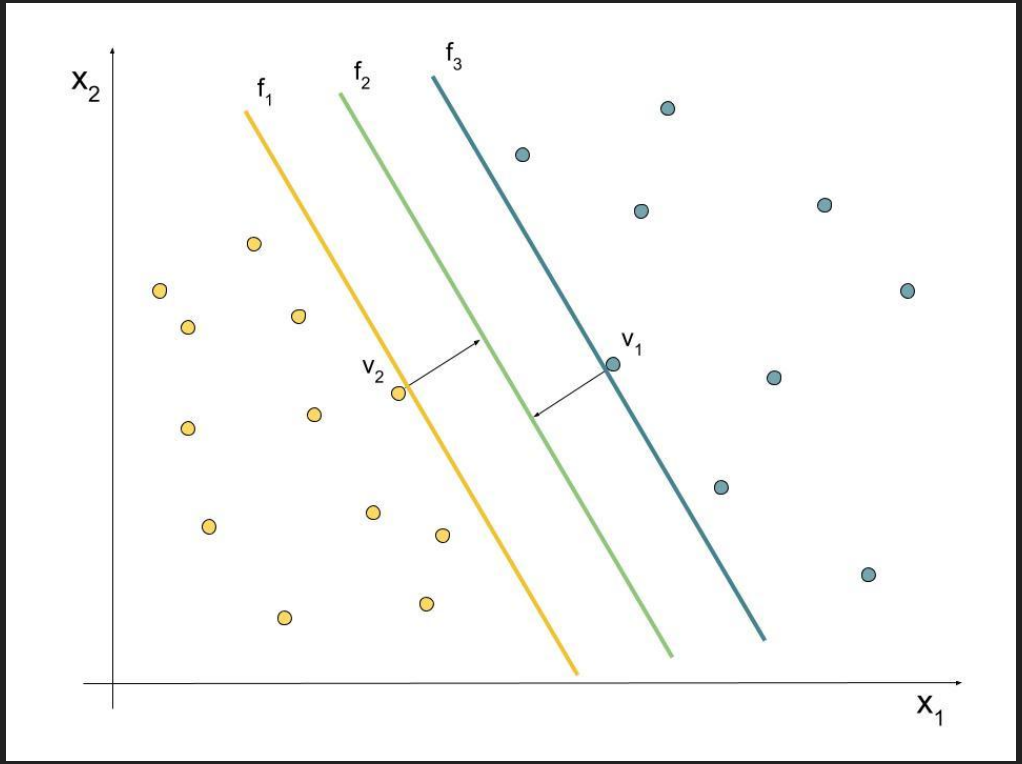
# Support Vector Machine

Es un clasificador que intenta hallar un hiperplano óptimo que separe 2 clases.

Utilizan lo que se denominan *kernels* para tratar problemas no linealmente separables, por lo que es altamente parametrizable.

En este trabajo se jugó a su vez con distintos valores para los parámetros C y gamma.

# Support Vector Machine



# Support Vector Machine - Parámetro C

En vista de que el criterio para determinar el margen es demasiado rígido e intenta clasificar perfectamente todos los puntos, el parámetro C define el costo de clasificarlos erróneamente, obteniendo un margen más suave, que puede fallar en algunos ejemplos, pero que da mejores resultados en el conjunto de datos en general y en ejemplos no observados en la etapa de entrenamiento.

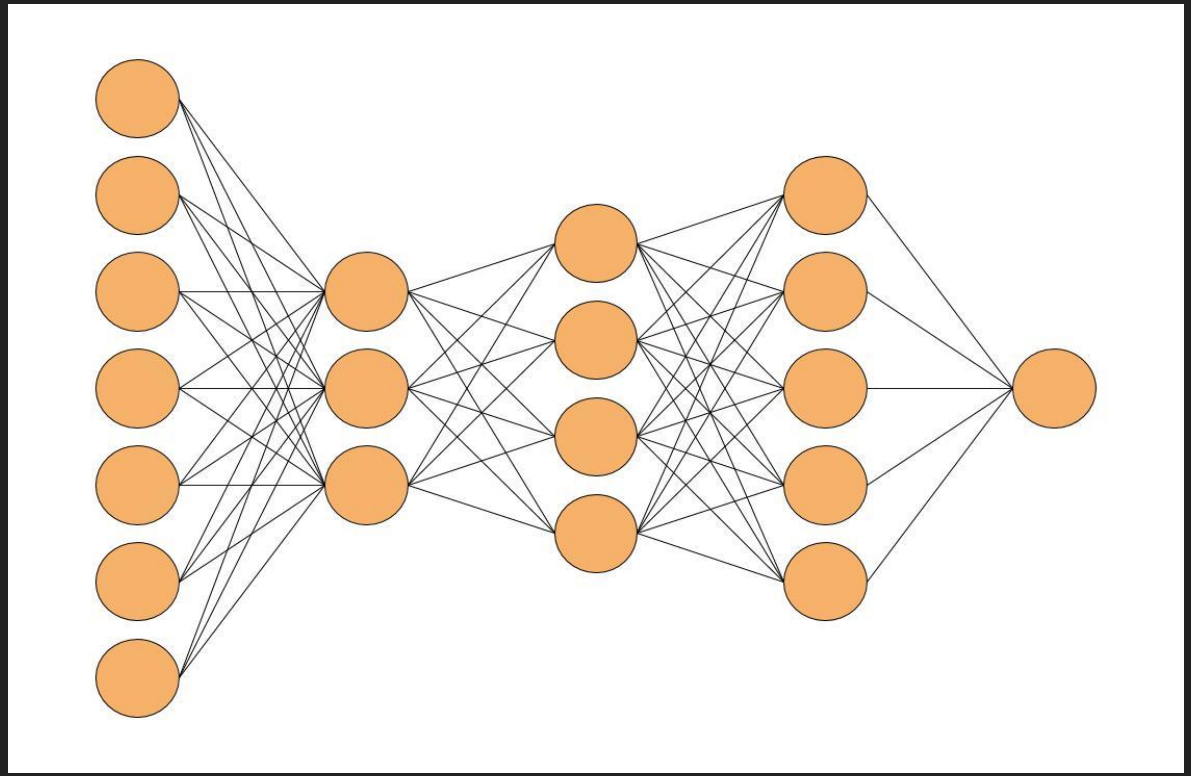
# Support Vector Machine - Parámetro Gamma

Gamma es utilizado como parámetro del kernel RBF, e indica el grado de influencia de los vectores de soporte sobre otros, es decir, un valor bajo de este parámetro puede tomar dos puntos muy lejanos como similares, mientras que un valor alto requiere que estos estén muy cerca.

# Multiperceptrón

El perceptrón multicapa, o multiperceptrón, es un tipo de red neuronal particular con una arquitectura feed forward, donde existe una capa de neuronas de entrada, una capa de salida y una o múltiples capas intermedias, denominadas capas ocultas. En este tipo de redes, la información se propaga desde la capa de entrada hasta la de salida, en donde se observan los resultados.

# Multiperceptrón



# Multiperceptrón

Los parámetros con los que se jugó en este modelo son: la función de activación, el algoritmo de optimización de la función de costo, la arquitectura de la red (cuántas capas ocultas y con cuántas neuronas cada una), y alfa (velocidad de aprendizaje).

# Estado del Arte

# Subjectivity Detection in Spoken and Written Conversations

Trabajo de Murray y Carenini

Se trabaja sobre un enfoque supervisado y uno no supervisado.

La base de datos consiste en correos electrónicos y transcripciones de reuniones en el enfoque supervisado, y artículos de blogs y noticias en el no supervisado.

Existen 3 clases de subjetividad: subjetivo-positivo, subjetivo-negativo y objetivo

# Subjectivity Detection in Spoken and Written Conversations

El método es muy interesante. Definen trigramas que representan patrones de términos en el discurso, por ejemplo: (very, ADJETIVO, ADVERBIO), y construyen un ranking de estos trigramas basándose en su probabilidad condicional P(es relevante / es x trigrama), contrastando textos con subjetividad opuesta.

En el enfoque no supervisado, los blogs se consideran subjetivos y los artículos de noticias objetivos, simplemente por la naturaleza de ambos textos.

# Subjectivity Detection in Spoken and Written Conversations

Junto con otras características crudas y conversacionales (pares de palabras que ocurren en la misma oración, pares de etiquetas POS que ocurren en una misma oración, dos tipos de longitudes de oración, etc), se entrena un clasificador de máxima entropía y se lo evalúa.

Las etapas en las que se divide la parte experimental son:

1. Detección de Oraciones Subjetivas
2. Detección de Oraciones y Preguntas Subjetivas
3. Detección de Oraciones Subjetivas-Positivas
4. Detección de Oraciones Subjetivas-Negativas

# Subjectivity Detection in Spoken and Written Conversations

Los mejores resultados fueron obtenidos en la etapa (2), demostrando que es más fácil detectar oraciones y preguntas subjetivas como un todo, que diferenciandolas en base a características más específicas, como la polaridad.

# Sentiment Analysis and Subjectivity

Trabajo de Liu. Es una recopilación del estado del arte en clasificación de sentimiento y subjetividad hasta la fecha en que fue escrito (2010).

Se divide en:

- El Problema del Análisis de Sentimiento
- Clasificación de Sentimiento y Subjetividad
- Análisis de Sentimiento Basado en Características
- Análisis de Sentimiento de Oraciones Comparativas
- Búsqueda y Recuperación de Opiniones
- Utilidad y Spam de Opiniones

# Sentiment Analysis and Subjectivity

La mayoría de los temas que trata se relacionan con el análisis de sentimiento, que si bien es distinto que el análisis de subjetividad, algunos métodos pueden aplicar en ambos casos.

En particular, en el ámbito de análisis de subjetividad se exploran los enfoques supervisado y no supervisado.

# Sentiment Analysis and Subjectivity

El enfoque supervisado consiste en experiencias con algoritmos de aprendizaje supervisado, donde se han identificado empíricamente algunas características que tienen mayor influencia que otras:

- Adjetivos y adverbios
- TF-IDF
- Etiquetas POS
- Negación
- Dependencia sintáctica
- otros

# Sentiment Analysis and Subjectivity

En el enfoque no supervisado, Liu describe el método de Peter D. Turney, donde se utilizan patrones de discurso subjetivos para extraer posibles oraciones subjetivas, se estima su polaridad y luego se examina si son efectivamente subjetivas, y en tal caso, se determina si contienen opiniones.

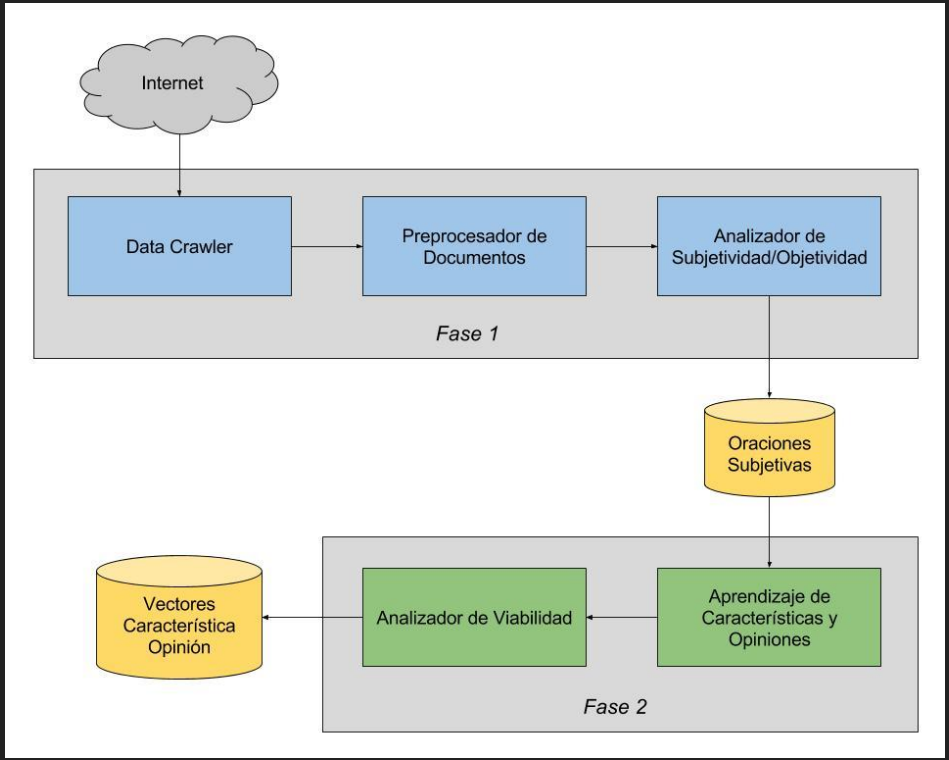
Para esta última etapa suelen utilizarse métodos supervisados, a veces combinados con métodos no supervisados.

# Subjectivity Classification using Machine Learning Techniques for Mining Feature-Opinion Pairs from Web Opinion Sources

Trabajo de Kamal. Consta de dos fases:

1. La primera fase utiliza un clasificador supervisado para determinar si las críticas de clientes a un producto dado son subjetivas u objetivas
2. En la segunda fase se analizan aquellas críticas subjetivas de forma semántica y lingüística a través de un método de reglas, a fin de obtener pares (característica, opinión) para cada producto.

# Subjectivity Classification using Machine Learning Techniques for Mining Feature-Opinion Pairs from Web Opinion Sources



# Subjectivity Classification using Machine Learning Techniques for Mining Feature-Opinion Pairs from Web Opinion Sources

1. Data Crawler: Extrae documentos de un sitio web.
2. Preprocesador de Documentos: Filtra porciones del texto según un análisis basado en etiquetas POS.
3. Analizador de Subjetividad/Objetividad: Por cada unigrama se crea un vector de características y se entrena un clasificador Naive Bayes que se utiliza para clasificar cada oración.
4. Aprendizaje de Características y Opiniones: Utiliza reglas y un análisis léxico para encontrar tuplas que identifiquen características y opiniones.
5. Analizador de Viabilidad: Determina la confiabilidad de cada tupla encontrada en la etapa anterior.

# Machine Learning approach for Subjectivity Classification based on Positional and Discourse Features

Trabajo de Chenlo y Losada.

Se propone un análisis del texto a nivel oración, basándose en la teoría de estructura retórica. A su vez, se proponen un conjunto de características que luego son utilizadas en dos modelos, un clasificador SVM y uno de regresión logística.

La evaluación se realizó en contraste con OpinionFinder

## Machine Learning approach for Subjectivity Classification based on Positional and Discourse Features

Utilizaron la versión en inglés de la base de datos NTCIR-7, que contiene noticias de diferentes fuentes y sobre diferentes temáticas.

Los modelos demostraron superar aquellos definidos en OpinionFinder, y se encontró que las características retóricas no jugaron un papel esencial por sí mismas en la detección de subjetividad, pero sí probaron ser de mucha utilidad para respaldar y dar fuerza a otras características, mejorando en gran medida los resultados finales.

# Método No Supervisado para la Detección de Subjetividad

Trabajo de Ortega Bueno. Realizado sobre textos en inglés.

Se propone un método no supervisado basado en oraciones, que introduce componentes de desambiguación semántica y análisis de sentidos.

# Método No Supervisado para la Detección de Subjetividad

Se evaluaron 3 técnicas diferentes de desambiguación semántica:

- Basada en grafos, donde los conceptos son vértices relacionados entre sí
- Basada en clustering, donde cada cluster es un sentido predefinido
- Basada en el algoritmo de Lesk, que analiza las n palabras más cercanas a un término

Para la clasificación de sentidos se utilizó un diccionario de sentidos construido con diferentes herramientas externas.

# Método No Supervisado para la Detección de Subjetividad

Finalmente se clasifica la subjetividad de acuerdo a un puntaje asignado en relación a un umbral estimado de forma empírica.

El uso de un mecanismo de desambiguación semántica particularmente probó ser de gran ayuda, influyendo fuertemente en el resultado final de la clasificación.

# Construcción de una Base de Datos

# Construcción de una Base de Datos

Imposible contar con una base de datos de oraciones anotadas de acuerdo a su subjetividad en español. En consecuencia, fue necesario construirla.

Si bien el resultado final no es una base de datos de alta calidad, fue suficiente para evaluar el método y determinar su utilidad.

# Construcción de una Base de Datos

La base de datos consta de 2000 oraciones. La mitad de las mismas son subjetivas, y la otra mitad objetivas.

El método de extracción de oraciones fue asistido por software, pero el anotado y la selección de ejemplos fue completamente manual, de forma de obtener la más alta precisión en el anotado, sacrificando volumen de datos.

Las oraciones fueron extraídas de papers del CACIC publicados entre el 2005 y 2013, y de libros de distribución gratuita del Proyecto Gutenberg

# Construcción de una Base de Datos

La base de datos obtenida es un archivo de texto plano, con las siguientes características:

- Los datos son texto plano, codificados en UTF-8
- Cada línea de texto está compuesta por un indicador de clase y una oración, separados por el carácter ‘@’, y en ese orden. Los indicadores de clase posibles son dos: S (oración subjetiva) y O (oración objetiva)
- Las oraciones aparecen tal como se ven en los textos fuente

# Diseño de una Solución

# Diseño de una Solución

Para la representación de los datos se utilizan 2 formatos: matricial y vectorial.

Una matriz oración es una representación numérica intermedia de una oración, es decir, es una representación no comparable que no se utiliza en el modelo final, sino como un paso previo para medir las oraciones en una escala más interna y detallada.

Un vector oración es una representación numérica final de una oración, es decir, es una representación comparable utilizada por el modelo final.

# Diseño de una Solución - Representación Matricial

En la representación matricial, cada fila representa una palabra (en el orden en el que aparecen en la oración), y cada columna una de las siguientes métricas:

- SWF-ISF
- Frecuencia Relativa Subjetiva (FRS): $f_{xs} = n_{xs}/n_s$. $n_{xs}$ es la cantidad de ocurrencias de x en oraciones subjetivas, y $n_s$ la cantidad total de palabras en oraciones subjetivas.
- Frecuencia Relativa Objetiva (FRO): el equivalente de FRS en oraciones objetivas.
- Modificador: 1 si la palabra es un adjetivo o un adverbio, 0 en caso contrario.

# Diseño de una Solución - Representación Matricial

| Palabra Original | SWF-ISF | FRS | FRO | Modificador |
|---|---|---|---|---|
| el | 0.16716 | 0.00678 | 0.00805 | 0 |
| aguileño | 0.01381 | 0.0002 | 0 | 1 |
| príncipe | 0.013 | 0.0002 | 0.00008 | 0 |
| gozaba | 0.013 | 0.0002 | 0.00008 | 0 |
| reputación | 0.0076 | 0.0001 | 0 | 0 |
| hiperbólica | 0.0076 | 0.0001 | 0 | 1 |

**Tabla 5.1:** Ejemplo de representación matricial de una oración simple

# Diseño de una Solución - Representación Vectorial

En la representación vectorial, cada vector cuenta con las siguientes métricas, extraídas de las matrices asociadas:

- Media de SWF-ISF Máximos: la media de los 3 mayores SWF-ISF.
- Media de FRS: la media de las FRS de la oración.
- Media de FRO: la media de las FRO de la oración.
- Frecuencia Relativa de FRS sobre FRO: FR de palabras cuya FRS es mayor a su FRO.
- Frecuencia Relativa de Modificadores (FRM): FR de adjetivos y adverbios.

# Diseño de una Solución - Representación Vectorial

- Frecuencia Relativa de Patrones de Bigramas Subjetivos (PABS): FR de bigramas que coinciden con alguno de los patrones de bigramas subjetivos preestablecidos.
- Frecuencia Relativa de Patrones de Trigramas Subjetivos (PATS): FR de trigramas que coinciden con alguno de los patrones de trigramas subjetivos preestablecidos.

# Diseño de una Solución - Representación Vectorial

Patrones de bigramas contemplados:

- (ADJ, SUST)
- (SUST, ADJ)
- (VERB, ADV)

# Diseño de una Solución - Representación Vectorial

Patrones de trigramas contemplados:

- (ADV, ADJ,  SUST)
- (SUST, ADV,  ADJ)
- (VERB, ADV,  ADV)
- (ADV,  ADV, VERB)

# Diseño de una Solución - Representación Vectorial

| Media de SWF-ISF Máximos | Media de FRS | Media de FRO | $FR_{FRS/FRO}$ | FRM | PABS | PATS |
|---|---|---|---|---|---|---|
| 0.06465 | 0.00126 | 0.00136 | 0.83333 | 0.33333 | 0.28571 | 0 |

**Tabla 5.2:** Ejemplo de representación vectorizada de una oración simple

# Diseño de una Solución - Preprocesamiento

La etapa de preprocesamiento se divide en 3 grandes fases:

- Fase de Formato
- Fase de Numerización
- Fase de Vectorización

# Diseño de una Solución - Preprocesamiento

Fase de Formato: se agregan metadatos y se filtra información irrelevante. Compuesto por:

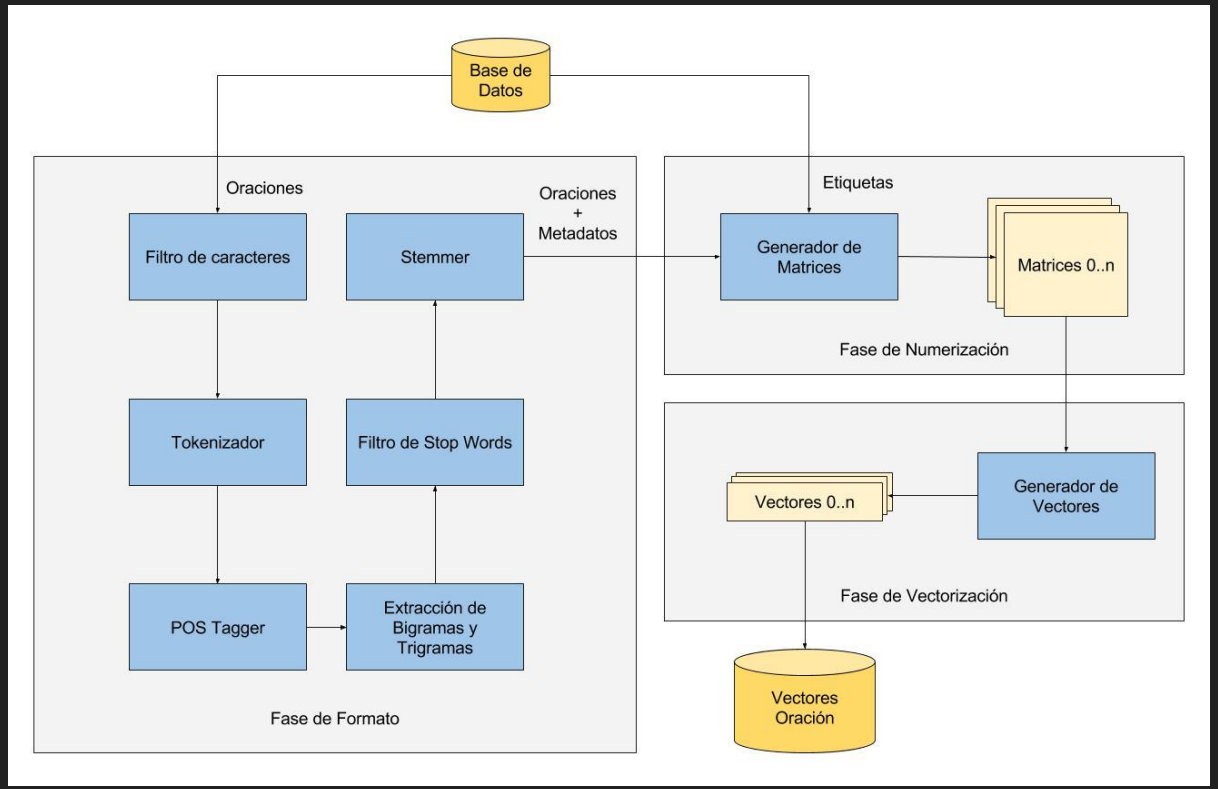
- Filtro de Caracteres: elimina caracteres de puntuación, signos de exclamación y dígitos.
- Tokenizador
- POS Tagger: utiliza el POS Tagger de Stanford para español, y obtiene una lista de pares (palabra, tag).
- Extracción de Bigramas y Trigramas
- Filtro de Stopwords: utiliza la lista de stopwords de NLTK para español.
- Stemmer: usa el algoritmo de stemming de Snowball para español.

# Diseño de una Solución - Preprocesamiento

Fase de Numerización: se recolecta toda la información adquirida en la fase de formato, se calculan las características deseadas y se construye la matriz asociada a cada oración.

Fase de Vectorización: en esta última etapa se toman las matrices generadas en la fase anterior, se calculan las características deseadas y se construyen los vectores oración asociados.

# Diseño de una Solución - Preprocesamiento



# Diseño de una Solución - Entrenamiento

Se entrenaron dos modelos, un SVM y un multiperceptrón, para los cuales se siguió un proceso de ajuste de parámetros y de optimización del entrenamiento en general, buscando las mejores particiones entre datos de entrenamiento y de prueba, todas en una relación 80%-20%

# Diseño de una Solución - Entrenamiento SVM

- 100 particiones de datos de entrenamiento y prueba
- 3 kernels: lineal, RBF y sigmoidal
- Diferentes valores de los parámetros C y gamma
- Validación cruzada con k = 5, buscando maximizar el f-score

El resultado óptimo para la partición óptima encontrada fue:

Kernel sigmoidal, C = 0.01, gamma = 0.001

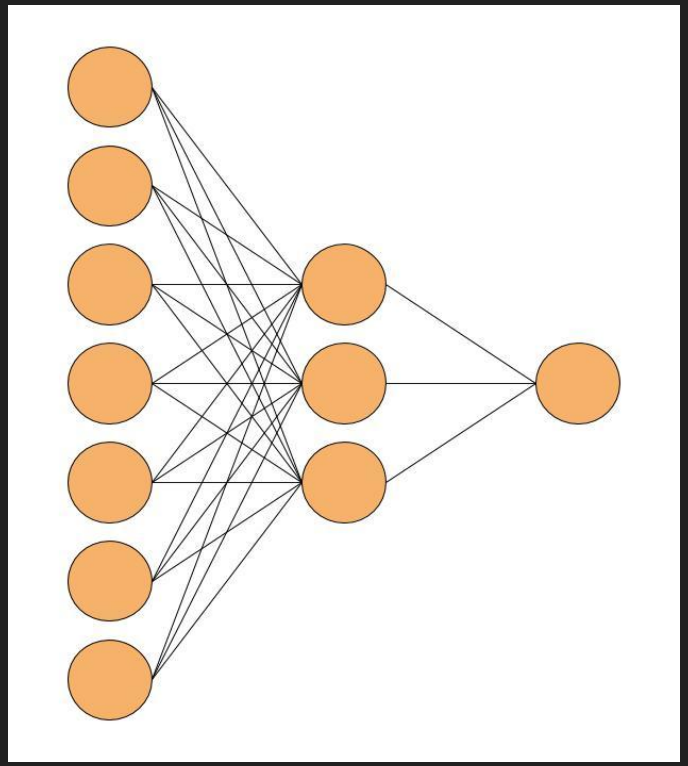
# Diseño de una Solución - Entrenamiento NN

- 10 particiones de datos de entrenamiento y prueba (~56.000 configuraciones cada una)
- Optimizadores libfgs y adam
- Funciones de activación logística, tangente hiperbólica y ReLU
- Diferentes valores de α
- Cantidad de capas ocultas y de neuronas en c/u de ellas
- Validación cruzada con k = 5, buscando maximizar el f-score

El resultado óptimo para la partición óptima encontrada fue:

Optimizador adam, activación ReLU, α = 0.01, 1 capa oculta con 3 neuronas

# Diseño de una Solución - Entrenamiento NN



# Conclusiones

La mayoría de las características de los vectores oración probaron no ser factores clave en la detección de subjetividad por sí mismos.

Sólo 2 características mostraron tener un impacto directo.

# Conclusiones

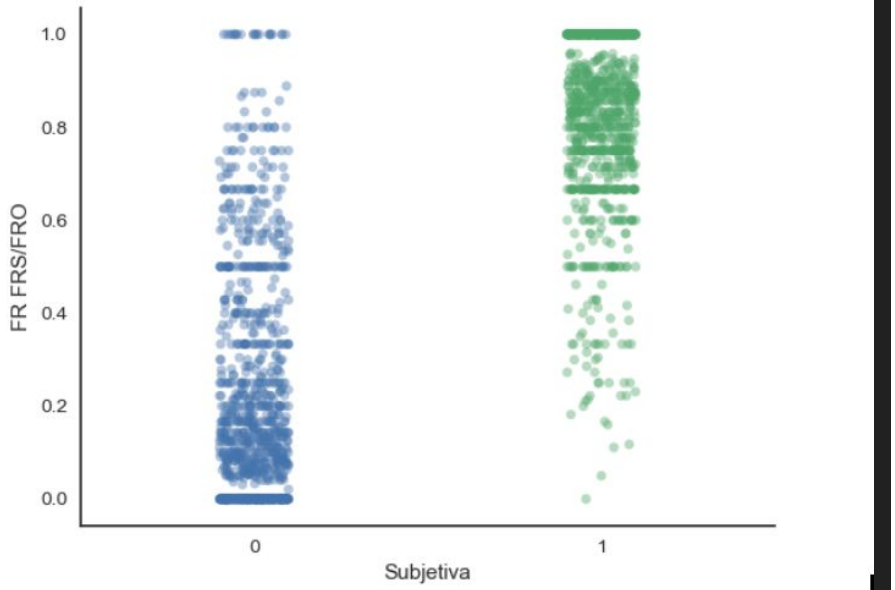



**Figura 7.7:** Relación entre frecuencia relativa de FRS/FRO y subjetividad

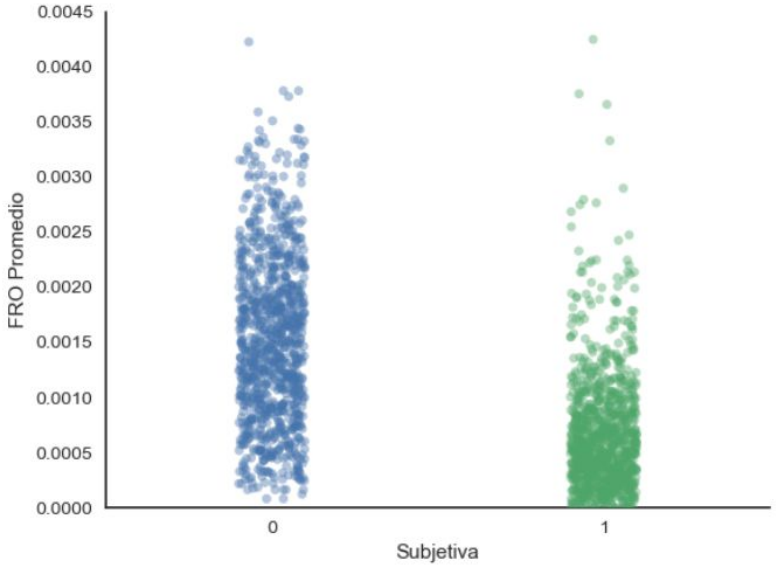



**Figura 7.8:** Relación entre FRO promedio y subjetividad

# Conclusiones

Uno de los SVM óptimos encontrados en la etapa de pruebas utilizaba un kernel lineal.

Gracias a esto se pudo extraer un gráfico indicando los coeficientes asignados a cada característica, indicando su importancia.

# Conclusiones

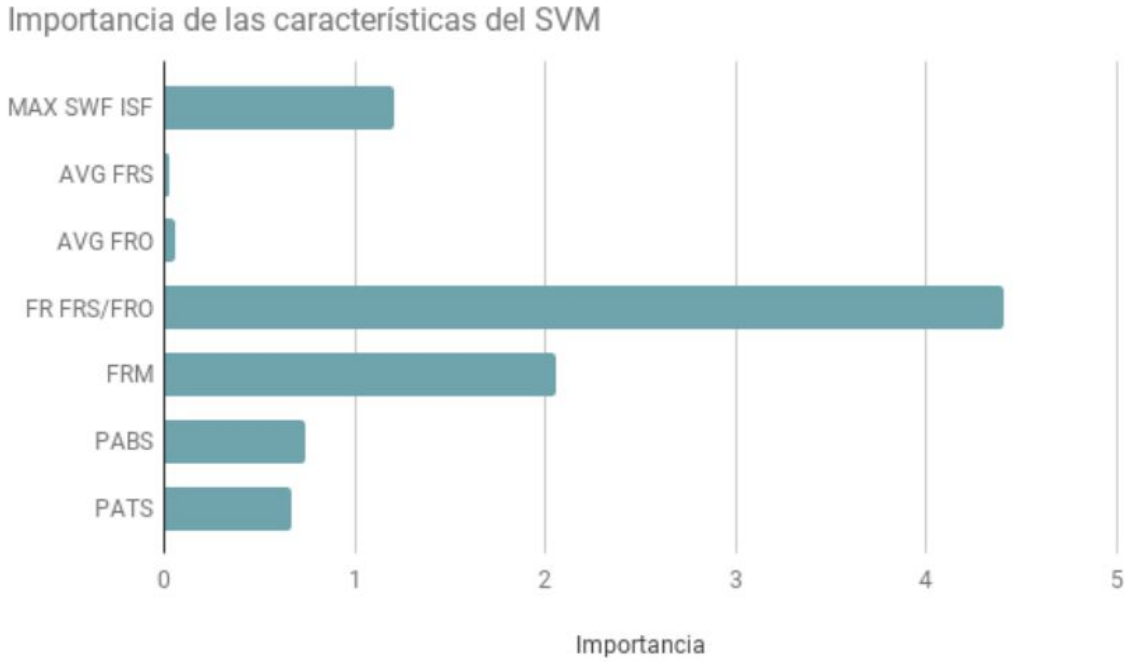



**Figura 7.9:** Importancia de las características del SVM

# Conclusiones

Ambos clasificadores fueron evaluados individualmente con sus datos de prueba, y con una validación cruzada con k = 5, repetida 30 veces.

Se calculó, entre otras métricas, el F-Score macro total promedio de la validación cruzada, y su desviación estándar.

# Conclusiones

SVM

- F-Score macro promedio: ~0.9
- Desviación estándar: 0.0003

Red Neuronal

- F-Score macro promedio: ~0.88
- Desviación estándar: 0.05

# Conclusiones

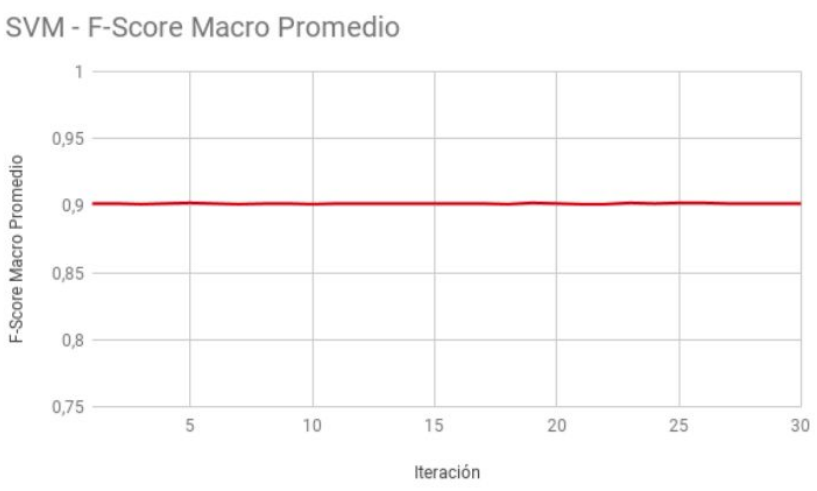



**Figura 7.10:** F-Scores Macro promedio del SVM

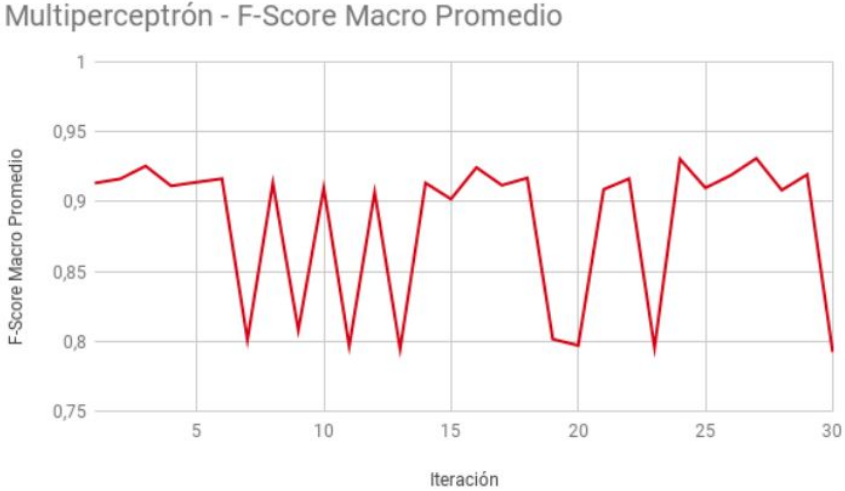



**Figura 7.11:** F-Scores Macro promedio del multiperceptrón

# Conclusiones

Se realizó una simulación de puesta en producción, utilizando la misma BD como entrada, y un estimador de subjetividad pobre de base para el SWF-ISF.

Se utilizó un estimador donde para cada palabra x de una oración se calcula:

$$S(x) = f_s(x)/f_o(x) \quad \text{si } f_o(x) <> 0$$

$$S(x) = 2 \quad \text{si } f_o(x) = 0 \text{ y } f_s(x) <> 0$$

$$S(x) = 0 \quad \text{si } f_o(x) = 0 \text{ y } f_s(x) = 0$$

Finalmente: si max(S) > 1, entonces es subjetiva, si no, es objetiva

# Conclusiones

La performance del estimador base apenas supera la clasificación aleatoria, con un 61.4% de predicciones correctas.

La sola aplicación del modelo eleva su performance en hasta un 30%, obteniendo un 79.8% de predicciones correctas.

Sería interesante observar los resultados producto de encadenar un clasificador conocido a un modelo de este tipo.

# Conclusiones

Otros posibles trabajos futuros:

- Mejorar la calidad y tamaño de la base de datos
- Explorar nuevas características
- Utilizar n-gramas de subjetividad específicos propuestos por profesionales de la lengua española
- Incluir algún mecanismo de desambiguación semántica en el pipeline de preprocesado

Muchas Gracias