Universitat Oberta de Catalunya

uoc.edu

Análisis de sentimientos en sistemas de ticketing IT

Jesús Navajas Briones
Máster Universitario en Ciencia de Datos (Data Science)
Text Mining & Social Network Analysis
(TM-8: Análisis de sentimiento en comunicación con
el departamento de IT)

Consultor/Profesor Asociado: Carlos Henrández Gañan
Responsable de la asignatura: Jordi Casas-Roma

Enero 2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Análisis de sentimientos en sistemas de ticketing IT*
Nombre del autor:	*Jesús Navajas Briones*
Nombre del consultor/a:	*Carlos Hernández Gañan*
Nombre del PRA:	*Jordi Casas-Roma*
Fecha de entrega (mm/aaaa):	01/2019
Titulación::	*Máster Universitario en Ciencia de Datos (Data Science)*
Área del Trabajo Final:	*TFM-Text mining & Social Network Analysis aula*
Idioma del trabajo:	*Castellano*
Palabras clave	*Sentiment análisis* *IT Tickets*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El estudio del análisis de sentimientos a textos (escritos o hablados) se ha aplicado en múltiples facetas distintas (escritos de redes sociales, interacciones con bots, evaluaciones o ratings de webs...); siendo el enfoque de este trabajo su aplicación en los sistemas de tickets o incidencias de IT.
El lenguaje e información suministrada en estos mensajes esta sujeta a su propio contexto (normalmente indicación de un error o problema) debiendo intentarse que este no influya en la evaluación final del sentimiento del emisor. Por ello se evaluarán diferentes técnicas para soslayar dicho problema. Principalmente se propone la construcción de un lexicon dedicado.

Abstract (in English, 250 words or less):

Sentiment Analysis (SA) has been widely proved in many areas: movies and accommodations reviews, product opinions, microblogs…The main point of this work is applying these technics to the messages of a IT's ticketing/incident management system.

The special characteristics of the messages involved, normally reporting an error or problem, force us to try to isolate the true emotion of the sender from this unwanted situation reported.

The first approach is building a domain oriented lexicon.

Índice

Lista de tablas y figuras

1. Propuesta

1.1 Propuesta de titulo

Se propone como titulo: *Análisis de sentimientos en sistemas de ticketing IT*

1.2 Palabras clave (keywords)

Se establecen dos palabras claves:

- *Sentiment Analysis*
- *Ticket IT*

1.3 Resumen de la propuesta (abstract)

Partiendo de la propuesta inicial:

El análisis del sentimiento se ha adoptado en algunos departamento de IT para problemas tales como congestiones de red y el sentimiento de usarios infectados. Este TFM pretende definir un método para evaluar el sentimiento contenido en los tickets para soporte de TI (Tecnología de la información). Los tickets TI tienen una amplia cobertura (por ejemplo, infraestructura, infecciones) e implican errores,incidentes, solicitudes, etc. El principal desafío es distinguir automáticamente entre la información fáctica, que es intrínsecamente negativa (por ejemplo, descripción del error), del sentimiento incrustado en la descripción. El enfoque del TFM sería crear automáticamente un diccionario de dominio que contenga términos con sentimiento en el contexto de TI, utilizado para filtrar los términos para el análisis de sentimiento.

Se plantea un abstract como el siguiente:

El estudio del análisis de sentimientos a textos (escritos o hablados) se ha aplicado en múltiples facetas distintas (escritos de redes sociales, interacciones con bots, evaluaciones o ratings de webs...); siendo el enfoque de este trabajo su aplicación en los sistemas de tickets o incidencias de IT.
El lenguaje e información suministrada en estos mensajes esta sujeta a su propio contexto (normalmente indicación de un error o problema) debiendo intentarse que este no influya en la evaluación final del sentimiento del emisor. Por ello se evaluarán diferentes técnicas para soslayar dicho problema. Principalmente se propone la construcción de un lexicon dedicado.

1.4 Descripción de la propuesta y justificación del interés y la relevancia de la propuesta

El trabajo se propone como un estudio comparativo entre técnicas ya establecidas de análisis de sentimientos en textos respecto a posibles mejoras que sean de aplicación dentro del ámbito de las incidencias IT.
Supone un ejercicio e investigación en si misma al plantearse distintos enfoques dentro del estado del arte del NLP actual; siendo a la vez de aplicación a posibles sistemas en funcionamiento al aportar información adicional para la priorización, evaluación o revisión de los procesos de actuación.

1.5 Explicación de la motivación personal

Dentro de las propuestas para la realización del TFM hemos escogido está por dos razones principales:

1. Poder aplicar distintas técnicas de NLP y hacer comparativas,
2. circunscribirse a un ámbito en el que tenemos amplía experiencia (llevo trabajando 20 años como Informatico y he interactuado mucho con este tipo de sistemas).

1.6 Definición de los objetivos (principales y secundarios)

Como objetivo principal del TFM se plantean la obtención de un modelo para la evaluación del sentimiento en los sistemas de ticketing que suponga una mejora respecto a técnicas generalistas ya aplicadas en otros ámbitos.

Secundariamente:

- Establecer procedimientos de mejora continua para dicho modelo.
- Realizar un análisis del estado del arte en diferentes técnicas.

1.7 Descripción de la metodología empleada en el desarrollo del proyecto

Al tratarse de un proyecto de Data Science y en el que evaluaremos la bondad de diferentes modelos se aplicarán métodos cuantitativos/estadisticos que refuten los diferentes modelos planteados.
Deberá establecerse conjuntos de datos de prueba y validación (a definir técnica: cross validation…) y contrastes de hipótesis estadísticos -cuando sean aplicables- para soportar la validez/medida de mejora de los modelos.

1.8 Planificación o plan de investigación del proyecto

Como plan de investigación se identifican las siguientes tareas hitos principales (basándose en la planificación de entregas ya establecida):

- FASE I (PEC 2):
    - Estudio del estado del arte de diferentes técnicas. Selección de técnicas a comparar y a desarrollar. En principio se plantea como primera mejora uso de un diccionario y lexicons, pero se podrán evaluar otros modelos basados en vector embedded u otros.
    - Selección inicial de herramientas a utilizar (lenguaje, librerias, pipelines, recursos de proceso…).
    - Delimitación de fuentes de datos disponibles a utilizar. Validación y cleansing de los mismos. El profesor dice ya disponer de un DataSet pero habrá que ver si está etiquetado, formato…

- FASE II (PEC3): Construcción y evaluación de los distintos modelos. Se planteará un enfoque progresivo, aplicando nuevos métodos o mejoras siempre que el tiempo lo permita.
- FASE III (PEC4): Documentación. Redacción de la memoria.

* Aplico una especie de metodología ágil, considerando cada una de las fases un sprint, y por ello he puesto más énfasis en definir el siguiente (FASE I)

1.9 Bibliografía

Adicionalmente a la bibliografia existente en Sentiment Analysis, en una primera búsqueda se ha encontrado solo una referencia directa:

- Publicación: 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories
- **Titulo: Sentiment Analysis in Tickets for IT Support**
- Autores:
    - Cássio Castaldi Araujo Blaz
      Instituto de Informática
      Universidade Federal do Rio Grande do Sul
      Porto Alegre, Brazil
      ccablaz@inf.ufrgs.br
    - Karin Becker
      Instituto de Informática
      Universidade Federal do Rio Grande do Sul
      Porto Alegre, Brazil
      karin.becker@inf.ufrgs.br

2. Estado del arte

2.1 Consideraciones generales

El planteamiento a seguir para realizar el Estado del arte será el siguiente:

- En primer lugar realizaremos un resumen de las distintas técnicas existentes de uso común dentro del ámbito del sentiment analysis. Plantearemos el problema conceptual y modelos de amplía extensión.

  En este punto utilizaremos como base uno de los textos de las asignaturas del propio master (1) ,dado que nos encontramos en fase de estudio de dicha asignatura y estimamos servirá como refuerzo para el aprendizaje. En la entrega final se evaluará la extensión de dicho apartado y su posible resumen. Dado que la finalidad de la tesis no es dicho estudio, no pretende ser una descripción detallada sino un punto de partida. Existen estudios propios detallados sobre esta comparativa (2).

- Segundo, expondremos el resultado de una investigación -usando principalmente internet – de nuevas tendencias dentro del campo de sentiment analysis.

- Por último, se ha focalizado la búsqueda dentro del análisis aplicado a gestión de incidencias y más específicamente en las incidencias TIC.

2.2 Estudio General

2.2.1 Planteamiento del problema

El sentiment analysis se engloba dentro de los problemas de clasificación de fuentes de lenguaje natural dentro del área de la subjetividad/afectividad. Es decir:

- Se trata de un problema de clasificación, para cada elemento de entrada debemos clasificarlo, o en algunos casos ofrecer un numero valorativo y umbrales (que puede ser la probabilidad de pertenencia a las diferentes clases). Normalmente el sentimiento de dividirá en positivo (a favor), negativo (en contra) o neutro.

- Los elementos a clasificar son documentos de lenguaje natural. Entenderemos normalmente textos escritos, pero también se aplica a lenguaje hablado. Será pues parte de lo que se denomina NLP (Natural Language Processing).

- Se trata de clasificar por la subjetividad del autor en el momento de su realización. No estamos tanto dentro del ámbito de la semántica pura -de qué trata el texto- sino de la afectividad respecto a lo que está tratando.

Englobándose dentro de lo que podemos entender por modelos de Data/Text Mining, Machine Learning o Artificial Intelligence, nuestra finalidad es construir un modelo que permita realizar dicha clasificación. Un modelo:

- Tendrá una serie de atributos (features) de entrada que lo alimentan. Normalmente expresados mediante números (reales si son cantidades continuas).
- Requiere de aprendizaje, que se clasifica en supervisado (si le aportamos un conjunto de ejemplos ya etiquetados) o no supervisado/ semi-supervisado si es capaz de aprender de alguna manera independiente a partir de unas semillas iniciales.
- Deberá evaluarse su bondad o efectividad a partir de un conjunto de pruebas o conjunto de test, estableciendo distintas métricas (accurancy, precisión, recall, ROC, f1…).

En el campo del Sentiment Analysis se ha usado ampliamente conjuntos de valoraciones ya etiquetadas en webs, tipo evaluación de libros o películas, existiendo múltiples dataset [1] y competiciones.

Se trata de una evaluación de afectividad o subjetividad, distinguiéndose estos análisis de aquellos puros de clasificación de contenido, autoría del texto, etiquetado como spam/no spam... Entramos en el ámbito de la computación afectiva y detección de sentimientos, relacionada con la psicología y recogida en amplia literatura (3); y distinguiéndose normalmente según la duración y relación del sentimiento en : emoción, sentimiento, personalidad, estado emocional y actitudes (emotion, sentiment, subjectivity,personality, mood, and attitudes). Si bien, como hemos ya comentado, en muchos caso se reducirá a una clasificación en tres : Positivo, Negativo o Neutro.

A la hora de definir cuales serán las entradas o atributos que alimentarán nuestro modelo, debemos tener en cuenta que partimos de un documento o conjunto de texto natural, lo cual abre muchísimo nuestro abanico de posibilidades, y obliga normalmente a un preproceso del mismo para que modelos de uso común en otros ámbitos sean de aplicación. El lenguaje tiene muchos elementos que pueden valorarse/medirse, desde las propias palabras conteo/frecuencia -valorando su semántica -, hasta las construcciones verbales (sintaxis: frases imperativas…) o uso de otros elementos (Exclamaciones, emoticonos…); viéndose además más enriquecido si se trata de texto hablado (tono, volumen…). Normalmente el preproceso incluirá un conjunto de los siguientes:

- Tokenización. Separación de elementos (palabras, signos de puntuación …). Incluyéndose en ciertos casos la detección de colocaciones: conjuntos de tokenes que se entienden como entidades únicas (nombres compuestos, verbos con partícula…)
- Tagueado de los tokens. Asignándole la categoría sintáctica a cada token (POS- Part of Speech): Verbo, sustantivo, adjetivo…

[1] SemEval 2013, SemEval 2014 (Rosenthal, 2014), Vader (Hutto, 2015), STS-Gold (Saif, Fernandez, He, & Alani, 2013), IMDB (Mass et al., 2011) and PL04 (Pang et al., 2002).

- Stematizacion o lemanización. Simplificación de tokens en formas únicas (eliminación de plurales, conjugaciones verbales…).
- Análisis sintáctico/gramatical de oraciones (parser).
- Deducción de dependencias semánticas.

Debe entenderse que los lenguajes humanos son gramáticas ambiguas, donde además existen palabras con múltiples significados, por lo que estos procesos son probabilísticos/heurísticos. Se complica el problema aún más si tenemos en cuenta que puede depender del idioma de origen y del dominio de conocimiento al que el documento se refiere.

Procederemos a evaluar los posibles atributos y diferentes modelos en más detalle.

2.2.2 Atributos

- **Selección de atributos**

El proceso de selección de atributos para un modelo -en sentido general- supone determinar el subconjunto, del total de atributos disponibles, que producen mejores resultados (o identificar aquellos que no aportan información para su eliminación). Debe entenderse que la mayor información será siempre aportada por el total de atributos, pero por limitaciones del modelo (correlación…), tipo de parámetros admitidos (discretos/continuos…), limitaciones en el tiempo de cálculo, convergencia a soluciones o simplemente para ampliar el conocimiento del problema planteado se intenta realizar esta reducción. Nótese que al tratarse de lenguaje natural el número de atributos posibles es enorme (pensemos por ejemplo que establecemos una dimensión por cada termino distinto, en muchos lenguajes esto supone millones).

Si se dispone de un corpus etiquetado se pueden aplicar contrates paramétricos de estadísticos o de ganancia de información (Information Gain) para prefiltrar aquellos atributos que se identifican como relevantes, si bien en muchos tipos de modelos (como veremos posteriormente) los parámetros obtenidos del aprendizaje de este nos informará de la relevancia de cada atributo de entrada (modelo explicativo). En otros casos se pueden aplicar técnicas iterativas, de inclusión de atributos/eliminación del conjunto, evaluando el comportamiento del modelo contra un conjunto de test en cada paso.

Pudiendo también aplicarse técnicas de reducción de la dimensionalidad tipo PCA o SVD.

Los atributos usados comúnmente para sentiment analysis incluyen:

- **Conjunto de palabras (Bag of words).**
  Se toma como atributo la aparición de ciertas palabras (bien como suma de apariciones (1/0) , por frecuencia relativa, suma de pesos…) en cada uno de los documentos.
  Para seleccionar las palabras puede optarse por:

- Partir de lexicons ya establecidos donde las palabras tienen asociado un peso positivo/negativo (seleccionando por umbral para un conteo SI/NO o promediando pesos).
- Utilizar métodos semi-supervisados para generar el conjunto de palabras a partir de un conjunto inicial. Se partirá de un conjunto de palabras positiva/negativas y se enriquecerá:
  - Identificando palabras similares a partir de conjunciones con aquellas ya identificadas (tipo XXX y XXXX o XXXX pero XXXX) en el corpus.
  - Obteniendo sinonimos/antonimos a partir de fuentes léxicas/thesausrus (tipo Wordnet, Conceptnet)
  - Por relaciones entre palabras aplicando n-grams(palabras que aparecen conjuntamente en ventanas de n y medidas tipo PMI).
- Obtenerlas a partir de una vectorización de documentos de un corpus ya etiquetado (que veremos en siguiente punto).

Entenderemos, en este caso, que los atributos de entrada a nuestro modelo lo constituye unos valores agregados calculados a partir del número de palabras en la bolsa seleccionada (suma de pesos, suma de apariciones, frecuencias relativa, suma de pesos ponderada…)

- **Vectorizacion de documentos.**

  En este caso se opta por identificar con cada documento un vector con el conteo o frecuencia relativa de las palabras que incluye. Para determinar el conjunto inicial de palabras a tener en cuenta puede usarse filtros:
  - por tipo de palabra (seleccionado solo sustantivos, verbos, adverbios…y eliminando palabras muy comunes -stopwords),
  - aquellos que aparezcan un mínimo de veces,
  - tengan un valor mínimo de alguna medida de información (tipo tf-idf) …

  En segundo lugar, si el corpus está etiquetado (se dispone de un conjunto donde ya esta asignada la clase de destino) puede aplicarse filtros adicionales usando test estadísticos (sobre las diferencias en las medias entre los conjuntos de cada clase) o métodos asistidos como el análisis de diagramas de Pott.

  Entenderemos que en este caso la entrada a nuestro modelo será el vector de frecuencias relativas o aparición (1/0) de cada palabra estimada como relevante. Pudiendo aplicarle procedimientos de reducción de dimensionalidad tipo SVD, en lo que se denomina LSA (Latent Semantic Analysis).

  Estas codificaciones suelen denominarse sparse dado que presentan muchos ceros en la matriz (son muchas palabras/dimensiones).

- **Vectorización densa de palabras (embedded).**

En los últimos años se utilizan técnicas que mapean palabras/tocken a un espacio vectorial real de dimensión establecida, de manera que las palabras similares (sintáctica/semánticamente) presentan cercanía (medida como similitud entre los vectores que las representan- distancia euclídea/coseno).

La obtención del mapeo entre palabras y vectores puede hacerse a partir del corpus de entrada, forzando la codificación de cada palabra en un vector de la dimensión establecida y entrenando el sistema mediante favorecer representaciones similares entre palabras cercanas (CBOW Continious bag of word) y/o predicción de palabras no presentes (skip grams). Si bien ya existen representaciones calculadas para conjuntos amplios entrenadas sobre corpus generalistas.

Como implementación mas usada esta el word2vec de Google, que dispone tanto de métodos de aprendizaje como una representación ya establecida en 300 dimensiones de cerca de 3000000 de términos (en ingles y otros idiomas).

El uso de esta representación de palabras (como vectores) permite alimentar a nuestro sistema con el conjunto de palabras (si limitamos tamaño de texto), alguna media ponderada de los vectores que la representan o el uso de agregados sobre proyecciones de vectores (que intentan captar alguna dimensión semántica relevante). Veremos en el punto de tendencias actuales otros posibles métodos.

Estas codificaciones se suelen denominar densas en contaprestación a las sparse anteriores.

Además de estos atributos de uso común, tal y como indicamos en el punto anterior puede enriquecerse nuestro sistema con otros atributos y/o preprocesos:

- Palabras en mayúsculas.
- Uso de construcciones gramaticales determinadas.
- Longitud de las palabras.
- Detección de negaciones.

…

Y debe entenderse que en las codificaciones indicadas se pueden usar además de palabras individuales combinaciones de palabras (colaciones/n-gramas).

2.2.3 Modelos/técnicas

Las técnicas o modelos más usados incluyen las siguientes (explicamos someramente cada uno de ellos):

- **Naive Bayes.**

Presupone una distribución estadística (variable aleatoria) para cada una de las clases a partir de los parámetros de entrada (normalmente multinomial si son

continuos los atributos). Estima los parámetros de dicha distribución por máxima verosimilitud a partir de los datos de la muestra etiquetada y se selecciona como clase asignada aquella que presenta mayor probabilidad.

- **Regresión logística.**

Se presupone que la probabilidad de pertenencia a una clase viene dada por la curva logística $(1/(1+ \exp-(b_0 + b_1 x_1 \ldots))$, donde las x son los atributos y b los parámetros a estimar.

Nuestro problema se convierte en una regresión lineal en el ambito de los logaritmos, que podemos resolver por mínimos cuadrados o métodos iterativos a partir de los datos etiquetados.

Normalmente en los modelos de sentiment analysys se etiqueta como 0 los negativos, 0.5 los neutos y 1 los positivos y se establecen posteriormente unos umbrales para asignar a cada clase.

Es un modelo explicativo ya que el peso asignado al parámetro de cada atributo nos dará una idea de su aportación (y en que sentido).

- **SVM Support Vector Machines**

Son clasificadores que particionar espacios vectoriales linealmente (mediante hiperplanos), de manera que optimizan la distancia entre las distintas clases. Aplicando diferentes kernels permiten la separación en particiones no lineales.

- **Arboles de clasificación**

Establecen un árbol de reglas que va dividiendo recursivamente los conjuntos a partir rangos en los valores de atributos de entrada. Finalmente, cada hoja se asigna a una de las clases de partida.

El método para la selección de variables/rangos suele basarse en la entropía de la clasificación obtenida (de manera greedy o voraz).

- **K-NN**

Establece la pertenencia a una clase a partir de las clases de elementos cercanos. Para ello los atributos de entrada deben ser un espacio con una métrica definida.

- **Redes neuronales**

A partir de la definición una unidad de construcción básica (percepton: con función de activación y pesos) se construyen redes conectadas que son entrenadas (asignando pesos a cada conexión).

Normalmente distinguimos entre FNN (Feed), redes sin bucles , RNN recurrentes y CNN convolutional (con elementos repetidos: filtros ,pooling …).

Actualmente se denomina Deep Learning a los modelos con redes neuronales que poseen capas ocultas (entre la entrada y la salida); y existe amplia

documentación sobre distintas funciones de activación (RELU, hiperbólica…), métodos de regularización (evitar sobrepesos:L2,DropOut), métodos de aprendizaje, …

Como en el caso de la selección de atributos, solo a partir de la prueba de cada modelo (que además normalmente tendrá parámetros adicionales) podremos deducir cual es de mayor aplicabilidad a nuestro caso. Debiendo además tener en cuenta requisitos de aprendizaje (disponibilidad de datos), convergencia de las soluciones, efectividad/rendimiento del modelo (por ejemplo el K-nn requeriía almacenar todos los elementos etiquetados y calcular la distancia a cada uno de ellos cada vez)…

Este documento (4) presenta una comparativa entre varios de los distintos modelos aplicados al sentiment analysis; y en el siguiente punto entraremos a evaluar algunos de ellos en más detalle como tendencia actual(sobre todo redes neuronales); pero a modo de resumen:

- Se utiliza Logistic Regresion o Naive Bayes para modelos sencillos en los que se desea obtener conocimiento del mismo (dado que la deducción de los parámetros nos explica la aportación de cada atributo). Han sido, conjuntamente con los arboles de decisión, los más usados históricamente (nótese que los arboles permiten -si se podan y simplifican- obtener un entendimiento del modelo obtenido).
- Como clasificadores lineales los SVM presentan un comportamiento superior.
- El método K-NN se utiliza cuando la división entre clases planteada por los demás no obtiene los resultados deseados (clases no separables linealmente).
- Las redes neuronales actualmente han superado a todos los demás métodos si bien pueden requerir de más datos para su aprendizaje y son modelos no explicativos. Debe entenderse que un Logistic Regresion es una red con un único nodo y ese tipo de activación.

2.3 Nuevas tendencias en sentiment analysis/NLP

En **primer lugar**, además de los modelos anteriores, se está dando gran énfasis a la construcción de modelos mixtos o combinaciones, lo que se denomina normalmente **Ensembles**. Las posibilidades son múltiples:

- desde usar reglas de mayoría para obtener resultado conjunto,
- utilizar las salidas de los modelos (combinándolo con atributos de entrada o no) para alimentar un nuevo modelo,
- utilizar técnicas de Bagging, Boosting o RandomForest para construir/agrupar conjuntos de modelos

En (5) se plantea una taxonomía de estos Ensembles y clasificación de atributos de entrada (clasico o de la superficie, embedings y de construcción especifica), para después ofrecer un resultado comparativo de estas distintas combinaciones. Por los resultados obtenidos, la combinación de distintos tipos

de atributos como entrada para un clasificador neuronal (Deep learning) obtiene los mejores resultados ($M_{SG+Bigramm}$ en la nomenclatura del articulo).

En **segundo lugar**, la especialización en **distintos dominios (twitter,** evaluación de marcas/productos…) ha ampliado el estudio de la importancia de los atributos escogidos (y métodos para los mismos) como podemos apreciar en (6) y (7).

En **tercer lugar**, se plantean distintas **arquitecturas** para la aplicación de **redes neuronales** al sentiment analysis de textos.

En su aplicación a frases o textos limitados han sido comunes las RNN con vectorizaciones densas dado que estas permiten replicar la estructura sin problemas de disipación del gradiente, como puede apreciarse en textos de referencia (8).

Si bien en los últimos años se ha estudiado la aplicación de redes convolucionales. Estas extraen atributos de frases o ventanas de conjunto de palabras que pueden ser agrupados mediante un maxpool y aplicados a una red neuronal única, lo que permite aplicarlos a textos de la longitud requerida; tal y como se comenta en (9) y (10).

Así como también el uso de denominadas RTNN (Recursive Neural Tensor Network), que parten de texto analizado (parsed) sintácticamente en forma de árbol y permiten evaluaciones que superan a aquellas que no tienen en cuenta la posición de las palabras (Bolsas de palabras); tal y como se indica en (11). Obteniendo los mejores resultados en frases únicas dado que es capaz de interpretar negaciones y construcciones complejas.

Por **último**, la aplicación de la codificación densa a párrafos, en lo que se denomina **doc2vec/sentence2vec/paragraph2vec.** Requiere de un corpus de aprendizaje y debe realizarse un aprendizaje por cada texto para la codificación (no supervisada), lo cual supone una carga de cálculo importante para evaluar cada nuevo elemento, pero presenta los mejores resultados hasta la actualidad para la clasificación de documentos/sentiment analysis , tal y como se explica en (12).

2.4 Gestión incidencias y TIC

Al buscar documentos relacionados con el sentiment analysis aplicado a la gestión de incidencias o de soporte IT hemos encontrado pocas entradas. Por ello partimos del documento (13) ya identificado en la primera parte de este documento y realizaremos un análisis en detalle:

- Identifica uno de los mayores problemas de la aplicación del sentiment analysis en la gestión de incidencias: al normalmente comunicarse un error o fallo en un sistema existen muchas palabras que por defecto son asignadas a sentimientos negativos que dentro de este dominio no tienen dicha consideración. Hay que separar el incidente o problema comunicado del sentimiento.

- Plantea la construcción de un diccionario de dominio especifico para el dominio en cuestión que sirva para aplicar un simple algoritmo de suma de pesos asociados a dichas palabras del dominio.
  Para construir dicho dominio utiliza un método semi supervisado:
  - Selecciona inicialmente un conjunto de terminos como semillas.
  - Utiliza WordNet para la detección de sinónimos.
  - Purga los sinónimos. Solo se mantendrán aquellos que tengan valoración en Sentinet, de cierto umbral y esta sea del mismo signo que la palabra original o semilla.

  Para el proceso se usará el par termino#etiqueta POS.
  Especial tratamiento requieren los n-gramas detectados y etiquetados inicialmente en cuyo caso se buscan combinaciones por sinónimos de palabras individuales.

- Se ha realizado el siguiente preprocesado del texto:
  - Traducción al español (se trataba de portugués/brasileño y las fuentes son poco fiables).
  - Tockenizacion.
  - Deteccion de n-grmas (hasta 3)
  - Tagueado POS.
  - Lemmanización.

  Este preproceso se realizo al corpus inicial para su análisis y será de aplicación a los diferentes ejemplos para la aplicación del modelo.

- Presenta mejoras utilizando lo siguiente:

  - Da especial tratamiento a los saludos iniciales y despedidas, dado que estos suelen ocurrir en estas comunicaciones (como en los correos) y pueden ser una importante fuente para construir atributos.
  - Utiliza las exclamaciones para dar mayor peso a los elementos que le precedan (doble si es una, triple si es más de una).

- Afirma haber obtenido importantes mejoras respecto al uso de métodos estándar (basados en SentiStrength/Sentinet) con su modelo de diccionario de dominio.

Adicionalmente, hemos encontrado una herramienta de gestión de tickets, que se comercializa como Saas, denominada teamsupport[2] y que utilizan el sentiment analysis.

Para ello cada texto de interacción es calificado de 0 a 100 en los siguiente sentimientos tipo:

- Positive emotions: Satisfied, Excited, Polite
- Negative emotions: Sad, Frustrated, Impolite, Sympathetic

[2] https://help.teamsupport.com/1/en/topic/ticket-sentiment-analysis

Afirma utilizar el motor de IBM Watson para realizar esta valoración de cada texto. Posteriormente suman la clasificación de 0 a 100 de cada una de las interacciones(+ positivas – negativas) y califica todo el proceso como positivo o negativo dependiendo de una formula lineal de dicha suma.

3. Estudio inicial: Análisis datos de origen

3.1 DataSet aportado

Como parte del TFM se aporta por parte de la UOC un dataSet formado por cuatro ficheros csv: generic_malware.csv, iot_malware.csv, MDNS_vulnerabiltiy.csv y resolver_vulnerability.csv.

Estos ficheros poseen tres columnas(atributos) cada uno denominados:

- User: Identificación de usuario ofuscado (texto)
- Ts. Dia y Hora (Timestamp) de cuando fue realizada la comunicación.
- Content: Contenido en texto de la comunicación realizada por el cliente hacía el sistema de gestión de incidencias.

Podemos entender que el conjunto de comunicaciones con un mismo user supone una conversación continua.

El texto ha sido traducido desde el holandés al inglés y se ha realizado una ofuscación para evitar la identificación de datos privados, sustituyéndose ciertos valores por tokens del tipo <num>, <address>, <email>… (lo trataremos en más detalle posteriormente).

3.2 Análisis descriptivo/cuantitativo

Procedemos en primer lugar a unificar todas las fuentes dentro de un único fichero (total.xls) añadiendo una columna origen con la denominación del fichero del que parte.
En total tenemos 3118 filas/registros/comunicaciones, con la siguiente distribución por orígenes

Fichero Origen	**Cuenta**
generic_malware	2026
iot_malware	430
MDNS_vulnerabiltiy	541
resolver_vulnerability	121
Total general	**3118**

Tabla 1: Distribución registros por origen

Pudiendo agruparse en 939 conversaciones (o usuarios) distintas.
Procedemos rápidamente a ver el tamaño de los contenidos aportados, por lo que procedemos a contar numero de caracteres y tokens (palabras + signos de puntuación) por cada mensaje/registro.

	mean	**std**	**Min**	**25%**	**50%**	**75%**	**max**
Caracteres	355.87	624.99	3.0	127.25	228.0	414.0	17253.0
Tokens	88.53	217.31	1.0	28.00	54.0	101.0	7168.0

Tabla 2:Longitud texto por entrada

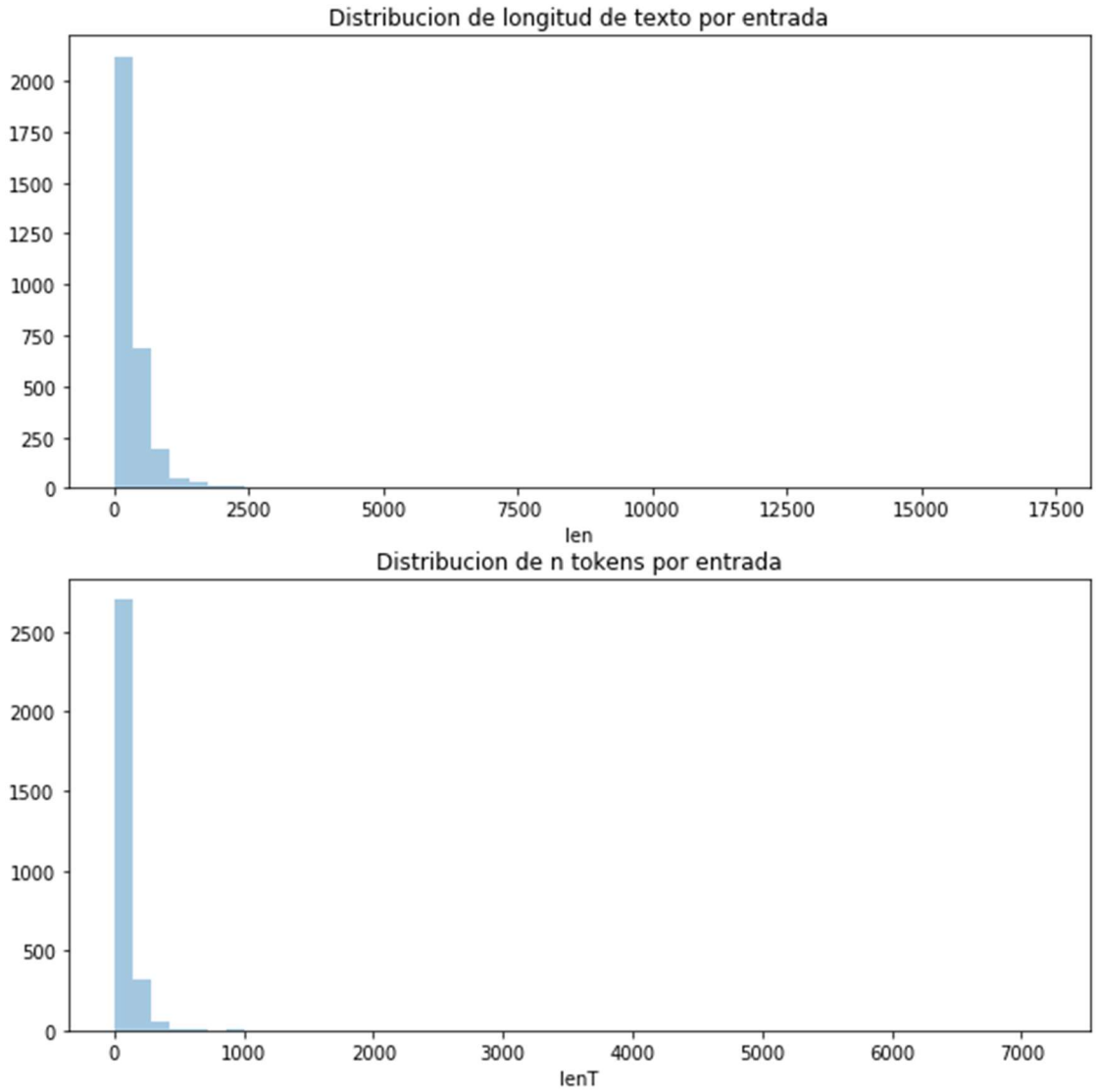

Ilustración 1: Histogramas longitudes texto por entrada

Donde apreciamos que los contenidos individuales son mayoritariamente pequeñas (<100 tokens <400 caracters).

3.3 Análisis del texto

Tras un análisis cuantitativo utilizamos herramientas automáticas de NLP para profundizar en la evaluación de los datos aportados.

En primer lugar, utilizamos el etiquetado automático de palabras/tokens dentro de las diferentes categorías gramaticales (sustantivos, verbos…) del contenido aportado, obteniendo la siguiente distribución:

ACRONIMO	TIPO	CONTEO	PORCENTAJE

J	ADJETIVO	16.236	6%
R	ADVERBIO	13.356	5%
N	NOMBRE	121.652	44%
V	VERBO	36.045	13%
X	OTROS	88.757	32%
		276.046	**100%**

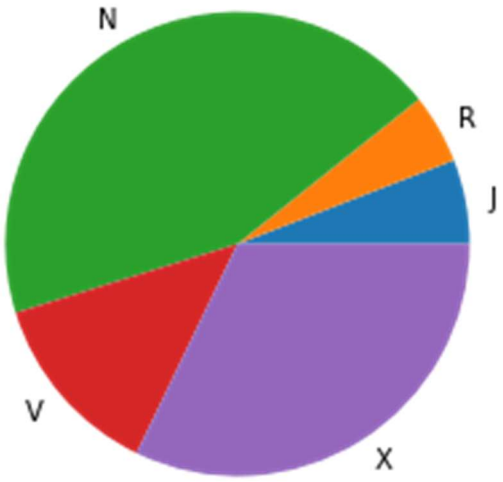

Ilustración 2:Tabla y Pie tipos gramaticales

Donde apreciamos que los nombres(sustantivos) son mayoritarios. Hay que entender que en ingles la diferenciación entre estas categorías no es evidente como en español y dependen del analizador (en este caso el de nltk que es secuencial y no un parser sintactico global).

En segundo lugar procedemos a identificar los datos privados enmascarados y su importancia en el contenido aportado. Para ello extraemos mediante expresiones regulares los tokens de la forma <palabra> del texto, obteniendo los siguientes resultados:

- Los términos ofuscados suponen casi un 10% del total del contenido, evaluando este como conjunto de palabras. Debe tenerse en cuenta que se han ofuscado todos los números y estos son muy usados (tanto para indicar teléfonos como valores relacionados con el propio problema transmitido: puertos, código de error…)
- Existen 27 distintos tokens usados para la ofuscación, si bien algunos parecen identificar el mismo elemento: `'<adress>', '<name>', '<num>', '<adddress>', '<file>', '<isp>', '<address>', '<addess>', '<website>', '<acompany>', '<technician>',`

```
'<customer>', '<technician>', '<helpdeskemp>', '<facebook>',
'<abusedeskemp>',    '<addrss>',    '<company>',    '<eum>',
'<adusedeskemp>', '<folder>', '<mail>', '<email>', '<summer>',
'<feed>', '<scan>', '<e-mail>'
```

Por último, se ha realizado un análisis cuantitativo de los términos más comunes, así como la identificación de colocaciones: bigramas y trigramas. Para los términos individuales se ha realizado un conteo puro, presentamos los valores más comunes detectados (top 10):

- ADJETIVO: ('dear', 911), ('good', 509), ('scan', 506), ('internet', 461), ('possible', 331), ('laptop', 305),('isp', 232), ('last', 213),('new', 198),('open', 190),
- ADVERBIO : ('also', 646), ('sincerely', 375),('still', 316),('already', 286),('soon', 234), ('back', 190),('longer', 152), ('anymore', 107),('completely', 101),('immediately', 97)
- NOMBRE: ('customer', 2408), ('num', 1890),('internet', 909),('connection', 880),('regard', 880), ('kind', 862),('problem', 666),('virus', 641),('computer', 558),('device', 556),
- VERBO: ('send', 782),('find', 496),('like', 424),('connect', 347),('get', 337),('see', 318),('use', 298),('remove', 273),('know', 257),('reset', 253),('scan', 231)

Donde apreciamos una clara relación con el ámbito del tema tratado.

Respecto a los bigramas y trigramas detectados, se ha realizado utilizando como medida el PMI (pointwise mutual information) , seleccionado aquellos con mayor medida de esta para todos así como para aquellos nominativos (constituidos de un nombre + adjetivo o dos nombres al menos). Se pueden consultar en los notebook jupyter aportados, pero indicamos que no se ha apreciado una casustica relevante y como tal – como veremos más adelante – no se incluirán como parte del análisis.

3.4 Procesamiento manual: Revisión y Etiquetado

La finalidad del trabajo es la construcción de modelos (basados en lexicons/BOW en primera aproximación) que permitan la clasificación automática de las comunicaciones textuales detectando aquella en las que el usuario/cliente no esta muy satisfecho (sentiment análisis). Pero, para poder aplicar métodos de aprendizaje supervisado, debemos partir de un conjunto etiquetado o ejemplos. Por lo que procedemos a una revisión manual de los datos aportados e intentar etiquetar como negativas aquellas entradas donde, subjetivamente, percibimos dicho sentimiento (seria mas adecuado haber dispuesto de un conjunto ya etiquetad mas contrastado, pero no ha sido posible).

Tras un trabajo de varios días llegamos a los siguientes resultados/impresiones:

- Tras la lectura de todos los mensajes, creemos que la calidad del texto es bastante pobre. Suponemos debido al propio contenido de este tipo de comunicaciones y su posterior traducción/ofuscación, siendo en muchos casos poco legible/comprensible.

- El número de elementos que han podido clasificarse como negativos han sido 85 de un total de 3118 (2.7%), lo cual creo nos da un conjunto muy reducido para un posible aprendizaje.

La poca calidad de los datos, junto a un volumen que podemos considerar reducido, creemos lastrará las posibles aplicaciones/construcciones de modelos, si bien se realizará la aplicación de distintos métodos y se verá el resultado obtenido. Por ejemplo, resulta casi imposible aplicar técnicas de Deep Learning con un conjunto tan reducido.

3.5 Primer modelo

Antes de embarcarnos en la selección de atributos(términos) y evaluación de modelos haremos una aplicación rápida de técnicas contrastadas para establecer una línea base u orden de magnitud de las bondades.

Para ello partiremos del conjunto de datos etiquetado y tokenizado y aplicaremos:

- Vectorización TFIDF (text frecuency/inverse document frecuency). Mediante esta técnica se asignará un vector a cada texto (mensaje) ligado a la frecuencia de cada termino dentro del elemento y ponderada inversamente por la frecuencia de aparición en los documentos. No limitaremos el número de términos, usando todos los términos del corpus (en torno a 5000).

- Clasificador de regresión logística usando los vectores antes expuestos y las clases dadas por el etiquetado realizado. Dado que nuestro conjunto esta claramente sesgado en cuanto a la ocurrencia de una clase u otra (las identificadas como negativas son el 2.7%), haremos que tengan el mismo peso ambos conjuntos (negativos y no identificados) .

- Para evaluar la bondad del modelo utilizaremos principalmente el recall y la precisión, dado que nos interesa saber que porcentaje de los negativos son detectados (recall) y de los indicados como tales cuantos realmente lo son (precisión). Nótese que la exhaustividad o accurancy en este caso puede ser muy engañoso, dado que al ser el más del 97% no identificados como negativos, un clasificador constante que diera siempre no negativo tendría dicha exhaustividad.

- Adicionalmente obtendremos las medidas anteriores de dos maneras:

    - En un primer caso utilizaremos el conjunto de datos para entrenamiento y mediremos también sobre el total (TEST=TRAIN).

    - En el segundo reservaremos un 20% del conjunto de datos para pruebas aprendiendo con el 80% restante (aplicaremos una selección estratificada, de manera que aseguremos elige un porcentaje similar de negativos en test y aprendizaje, relevante dado el sesgo de cardinalidad entre clases). Esto nos permitirá evaluar mejor la generalización evitando el overfitting.

Los valores obtenidos son los siguientes:

	TEST!=TRAIN	TEST=TRAIN
Accurancy	0,955	0,971
Recall	0,310	0,489
Precision	0,529	1,000

Tabla 3: Bondad primer modelo

Donde apreciamos que el accurancy es alto (dado el porcentaje de no negativos) y que en el caso total podemos llegar como mucho a un 50% de detección de las negativas usando este modelo (recall).

4. Selección de términos/features

4.1 Consideraciones generales

A partir del DataSet aportado plantearemos la construcción y enriquecimiento de un lexicón que pueda permitir la clasificación de los textos/mensajes a partir del conteo de ciertas palabras recogidas en un diccionario. Este enfoque, que ha sido el más utilizado en sentiment analysis (como indicábamos en el estudio del estado del arte), nos serivira para aboradar diferentes problemas de NLP, selección de atributos…

Como primera fase de dicho trabajo debemos seleccionar una serie de términos que sirva de semilla, lo cual tiene una serie de implicaciones:

- Debemos determinar que entendemos por termino/atributo (que palabras, signos de puntuación, lexemas…) y como esto se reflejara como feature/atributo.
- Deberemos seleccionar del conjunto total de términos/atributos aquellos que consideremos más relevantes para la tarea de clasificación que queremos abordar.
- Deberemos seleccionar un peso/medida asociado a cada atributo y un método de evaluación (función, suma…)

Aunque iremos tratando estos temas en los puntos sucesivos debemos entender que:

- No usaremos análisis de n-grams ni parser sintácticos. Creemos que los datos aportados no lo permiten, por lo que nos restringiremos a una tokenización y lemanizacion (para reducir combinatoria: plurales/tiempos verbales…)
- Como estimamos que nuestro clasificador se intentará aplicar a otros conjuntos de incidencias y debe ser lo más generalista, huiremos de vectorizaciones basadas en el corpus (tipo TDIDF) y supondremos simplemente un conteo del número de términos, identificándolo con las implementaciones más sencillas originarias del BOW (Bag of words).
- Para evaluar la bondad de nuestra selección de términos y para la asignación de pesos a cada palabra utilizaremos una regresión logística. Entendemos que aunque los métodos de selección de atributos pueden aportar valores numéricos (tipo Information Gain o p-value), la medición exterior aplicando un modelo de clasificación nos servirá para refutar los resultados (además de aportar una estimación para los coeficientes de los términos).

Usaremos distintas combinatorias y las compararemos al estilo de una grid de hiperparametros aplicadas a un modelo.

4.2 Procesado del texto

Plantemos, para la realización de estas tareas, el desarrollo de una serie de clases que actúan como filtros y puedan usarse a la manera de pipeline/DAG

para la obtención del resultado deseado (permitiendo la reutilización de resultados parciales para mejorar el rendimiento). Hemos realizado las siguientes:

- Preprocesar.
    - Elimina/sustituye las palabras ofuscadas originariamente.
    - Pasa a minúsculas todo el texto.
    - Tokeniza, separando los distintos elementos (términos, signos de puntuación)
    - Elimina duplicados de signos de puntación (stripear)
- PosFilter.Asigna a cada termino su etiqueta de POS (part of speech o categoría gramatical).
- StopLenFilter. Elimina palabras sin aportación semántica y repititivas (stop_words) y aquellas de longitud inferior a 3.
- LEM1Filter. Lemaniza las palabras sin tener en cuenta su POS.
- LEM1Filter. Lemaniza las palabras con su POS y distingue por el mismo.

La idea es parir del texto original y aplicar las siguientes transformaciones:

- Preprocesar-> PosFilter->StopLenFilter->LEM1Filter
- Preprocesar-> PosFilter->StopLenFilter->LEM2Filter

De esta forma obtendremos dos conjuntos de datos, uno en que los términos se identifican simplemente por su lema y otro por la conjunción de su lema y su función (POS). Denominaremos a estos conjuntos LEM1 y LEM2.
Si tomamos como ejemplo el primer mensaje:
Dear sir, Mrs., I have scanned my computer several times with Sophos, but can not open its details so that I can not view the view log file to forward the log file to you.What do I need to do to forward the log file to you? mvg, <Customer> <Email>

Quedaría reflejado de la siguiente forma:

- LEM1: ['dear', 'sir', 'mr', 'scanned', 'computer', 'several', 'time', 'sophos', 'open', 'detail', 'view', 'view', 'log', 'file', 'forward', 'log', 'file', 'you.what', 'need', 'forward', 'log', 'file', 'mvg', 'customer']

- LEM2: ['dear__J', 'sir__N', 'mr__N', 'scan__V', 'computer__N', 'several__J', 'time__N', 'sophos__N', 'open__V', 'detail__N', 'view__V', 'view__N', 'log__N', 'file__N', 'forward__V', 'log__N', 'file__N', 'you.what__V', 'need__V', 'forward__V', 'log__N', 'file__N', 'mvg__V', 'customer__N'

Posteriormente añadimos una LEM3 que solo tenía en cuenta elementos calificados como adjetivos o adverbios.

4.3 Selección de términos/atributos

Para seleccionar los términos más relevantes aplicaremos en primer lugar una vectorización a los textos de tipo contador: cada documento se identificará con un vector del conteo del conjunto de términos en el corpus.

Para posteriormente, una vez obtenidas dimensiones numéricas, obtener mediante dos técnicas estadísticas cuales de estas dimensiones (que terminos contados) son de mayor utilidad para nuestro problema de clasificación. Optaremos por usar:

- Chi2 (Método paramétrico) A partir de la media del conteo en los elementos pertenecientes a cada clase (negativos y no asignados) podemos aplicar un contraste estadístico tipo chi2.

- Mutual info (No paramétrico). Medida ligada a la entropía que nos indica la ganancia de información de la variable considerada (conteo de termino) sobre la objetivo (clasificador) [3].

Para cada uno de los selectores obtendremos las n mejores dimensiones/términos a tener en cuenta, para : 25, 50 , 100 , 250 , 500 y 2000 y observaremos estadísticas de lo obtenido (lo compararemos también con la opción de tener en cuenta todos los términos).

Como ejemplo mostramos los 25 términos/atributos seleccionados como más relevantes para el conjunto LEM2 usando el selector de mutual_info.

['acceptable__J', 'additional__J', 'availability__N', 'business__N', 'claim__N', 'confidence__N', 'damage__N', 'forward__R', 'future__N', 'hassle__N', 'hour__N', 'important__J', 'internet__N', 'limitation__N', 'pay__V', 'problem__N', 'provider__N', 'real__J', 'really__R', 'ridiculous__J', 'service__N', 'think__V', 'time__N', 'unlikely__J', 'working__N']

Debe entenderse que la presencia de estos términos se considera relevante para la clasificación, pudiendo ser favorables o desfavorables (pesos o coeficiente positivo o negativo para su señalización como de opinión/sentimiento negativo).

En la siguiente sección veremos un estudio comparado de estos conjuntos obtenidos.

4.4 Estudio de conjuntos de atributos/términos

Para combinación de conjunto de términos/preproceso (LEM1,LEM2), método de selección/selector (chi2 o multual_info) y numero de términos seleccionado (k =25, 50 , 100 , 250 , 500 , 2000 y total) hemos calculado una serie de estadísticas que ahora compararemos (se ofrecen todas en fichero Excel).

Primero analizaremos la media de términos por registro/entrada (para todas y solo para aquellas calificadas de negativas):

[3] https://en.wikipedia.org/wiki/Mutual_information

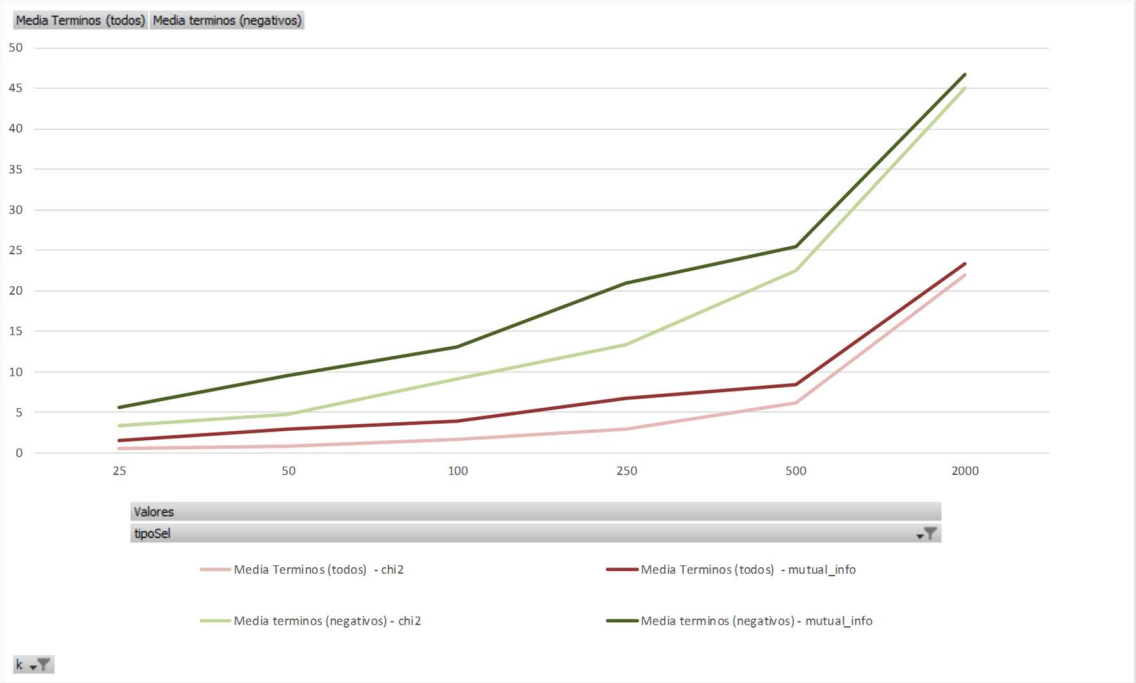

Ilustración 3:Evolución media de terminos

Donde apreciamos lo siguiente que la media de términos por entrada:

- Aumenta según aumentamos el número de términos tenidos en cuenta (lo cual parece lógico).
- Es mucho mayor para el conjunto de entradas etiquetadas como negativas (tono rojos) que para el total (verdes). Nuestros procesos de selección tomas como diferenciadores positivos mayoritariamente términos en dichos elementos.
- Es superior para términos seleccionados por mutual_info (serie mas oscura) respecto a la selección por chi2.

En las siguientes tablas presentamos estadísticos principales del número de términos por fila para todas:

preproceso	tipoSel	k	filas_mean	filas_std	filas_min	filas_25%	filas_50%	filas_75%	filas_max
LEM1	total	4704	26,07	33,88	0	10	18	31	726
LEM2	total	5311	25,05	32,69	0	10	17	30	721
LEM3	total	1536	5,05	7,70	0	1	3	6	174
LEM1	chi2	25	0,44	2,41	0	0	0	1	123
LEM1	chi2	50	1,22	3,31	0	0	1	2	154
LEM1	chi2	100	2,41	5,80	0	0	1	3	247
LEM1	chi2	250	3,37	6,80	0	1	2	4	256
LEM1	chi2	500	7,27	11,21	0	2	4	9	340
LEM1	chi2	2000	22,59	29,40	0	9	15	27	634
LEM1	mutual_info	25	2,08	3,21	0	0	1	3	52
LEM1	mutual_info	50	3,90	4,95	0	1	3	5	96
LEM1	mutual_info	100	5,46	7,03	0	2	4	7	157
LEM1	mutual_info	250	8,92	11,71	0	3	6	11	336
LEM1	mutual_info	500	10,74	13,72	0	4	7	13	348

LEM1	mutual_info	2000	23,95	28,38	0	9	16	28	459
LEM2	chi2	25	1,00	2,75	0	0	0	1	128
LEM2	chi2	50	1,06	3,08	0	0	1	1	145
LEM2	chi2	100	2,04	5,18	0	0	1	3	231
LEM2	chi2	250	2,81	6,07	0	0	2	3	249
LEM2	chi2	500	6,95	10,69	0	2	4	9	329
LEM2	chi2	2000	21,19	27,85	0	8	14	25	630
LEM2	mutual_info	25	1,64	2,66	0	0	1	2	49
LEM2	mutual_info	50	3,86	4,90	0	1	3	5	112
LEM2	mutual_info	100	4,95	6,30	0	2	3	6	143
LEM2	mutual_info	250	8,36	10,91	0	3	6	10	326
LEM2	mutual_info	500	10,19	12,84	0	4	7	12	339
LEM2	mutual_info	2000	22,62	26,62	0	9	16	27	426
LEM3	chi2	25	0,13	1,20	0	0	0	0	61
LEM3	chi2	50	0,16	1,24	0	0	0	0	61
LEM3	chi2	100	0,68	1,76	0	0	0	1	62
LEM3	chi2	250	2,62	4,01	0	0	2	3	78
LEM3	chi2	500	4,09	6,62	0	1	2	5	161
LEM3	mutual_info	25	0,97	1,62	0	0	0	1	26
LEM3	mutual_info	50	1,21	2,22	0	0	1	2	62
LEM3	mutual_info	100	1,24	2,39	0	0	1	2	74
LEM3	mutual_info	250	2,89	4,39	0	1	2	4	78
LEM3	mutual_info	500	4,30	6,29	0	1	3	5	135

Y solo para aquellas etiquetadas como negativas:

preproceso	**tipoSel**	**k**	**neg_filas_mean**	**neg_filas_std**	**neg_filas_min**	**neg_filas_25%**	**neg_filas_50%**	**neg_filas_75%**	**neg_filas_max**
LEM1	total	4704	48,51	51,54	4	20	33	54	375
LEM2	total	5311	46,19	49,65	4	18	31	51	365
LEM3	total	1536	9,78	11,38	0	3	7	10	80
LEM1	chi2	25	3,49	13,33	0	0	2	3	123
LEM1	chi2	50	6,39	16,97	0	2	3	6	154
LEM1	chi2	100	12,07	27,63	0	4	7	11	247
LEM1	chi2	250	16,67	29,67	0	7	11	16	256
LEM1	chi2	500	29,75	42,13	2	12	19	29	340
LEM1	chi2	2000	46,21	50,46	4	19	32	51	370
LEM1	mutual_info	25	7,35	7,43	0	3	5	10	51
LEM1	mutual_info	50	12,12	13,91	0	5	9	13	96
LEM1	mutual_info	100	17,62	22,12	0	8	12	19	157
LEM1	mutual_info	250	27,99	41,09	2	12	18	29	336
LEM1	mutual_info	500	34,20	45,03	2	14	24	34	348

LEM1	mutual_info	2000	47,84	51,23	4	20	33	54	373
LEM2	chi2	25	4,94	13,95	0	1	3	5	128
LEM2	chi2	50	5,85	15,98	0	1	3	6	145
LEM2	chi2	100	10,94	26,00	0	3	6	11	231
LEM2	chi2	250	14,85	28,57	0	5	9	14	249
LEM2	chi2	500	28,54	40,99	2	11	19	27	329
LEM2	chi2	2000	43,79	48,22	4	18	30	49	357
LEM2	mutual_info	25	6,34	7,02	0	2	4	9	49
LEM2	mutual_info	50	12,01	15,06	0	5	8	14	112
LEM2	mutual_info	100	16,33	20,28	0	7	10	19	143
LEM2	mutual_info	250	26,55	39,85	1	11	16	26	326
LEM2	mutual_info	500	32,75	43,70	2	13	21	35	339
LEM2	mutual_info	2000	45,49	49,41	4	18	31	51	363
LEM3	chi2	25	1,64	6,77	0	0	0	1	61
LEM3	chi2	50	2,04	6,86	0	0	1	2	61
LEM3	chi2	100	4,19	7,59	0	1	2	4	62
LEM3	chi2	250	8,34	10,75	0	3	6	9	78
LEM3	chi2	500	9,32	11,14	0	3	6	10	79
LEM3	mutual_info	25	3,22	4,40	0	1	2	4	23
LEM3	mutual_info	50	4,58	7,84	0	1	3	5	62
LEM3	mutual_info	100	5,40	9,14	0	2	3	6	74
LEM3	mutual_info	250	8,49	10,88	0	3	6	9	77
LEM3	mutual_info	500	9,58	11,30	0	3	7	10	80

Tabla 4:Distribucion nº terminos por fila

En segundo lugar veremos una comparativa de la evolución del recall de un modelo logístico aplicado sobre la vectorización restringida a los términos seleccionados para cada tipo de selección (chi2 y mutual_info) y Lemanización (LEM1 , LEM2 y LEM3):

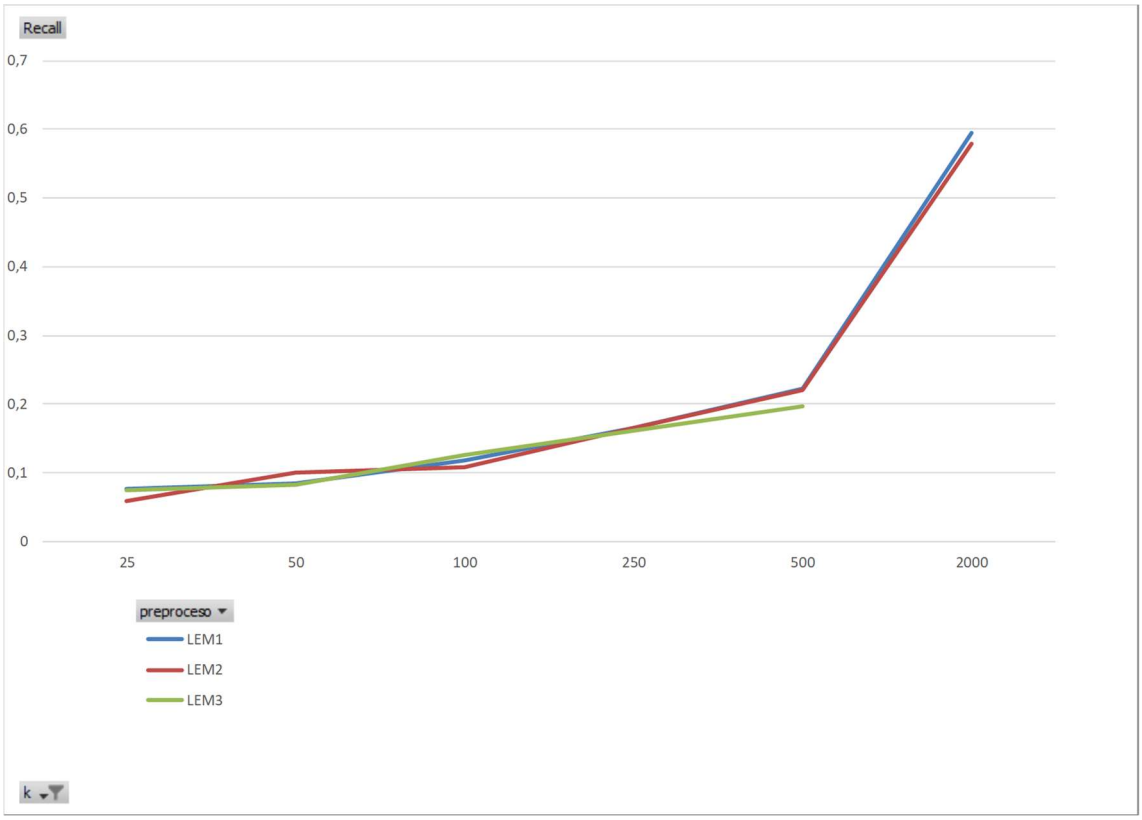

Ilustración 4:evolucion recall (por tipo preproceso)

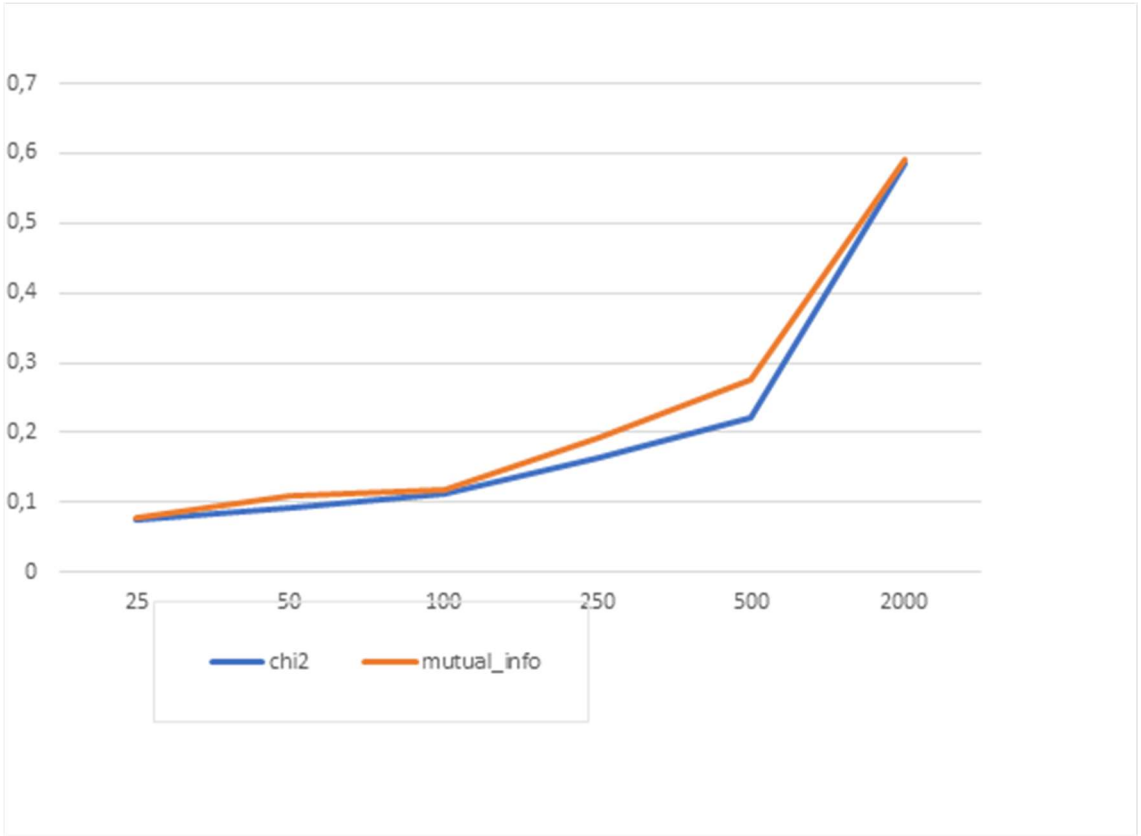

Ilustración 5:Evolucion recall (por tipo de seleccion)

Donde no se aprecía una diferencia significativa entre un preproceso y otro (si bien es ligeramente superior el LEM1) ; pero si parecen comportarse mejor los

conjuntos seleccionados mediante mutual_info respecto a los de chi2(lo cual parece coherente con que la media de términos sea superior).
Nótese que este recall es sobre el conjunto total, usado también como entrenamiento, en los datos adicionales se presenta los valores con un 80% para aprendizaje un 20% para test.

La diferencia de comportamiento entre ambos selectores (chi2 y mutual_info), nos lleva a estudiar el grado de similitud de los conjuntos seleccionados (intersección de ambos conjuntos evaluadas como porcentaje):

K	Porcentaje comun chi2 y mutual_info para LEM1
25	32%
50	38%
100	62%
250	54%
500	87%
2000	91%

Tabla 5: Porcentaje coincidencia métodos selección

Donde apreciamos crece pero es muy dispar en pequeños conjuntos

4.5 Asignación inicial de pesos

Tal y como hemos hicimos en el primer modelo hemos optado por construir un modelo de regresión logística para cada combinación con las siguientes características:

- Dado que nuestro conjunto de datos la clase etiquetada como negativa es de una cardinalidad muy reducida aplicamos una ponderación a dichos casos para que ambas clases tengan el mismo peso.
- Forzamos a que no use termino independiente (constante) de manera que el 0 de la multiplicación del conteo de términos (nuestro vector) por los coeficientes deba ser positivo para casos etiquetados como negativos (en sentimiento) y viceversa.

Entenderemos que los valores de los coeficientes nos podrán servir para implementar pesos en nuestro clasificador dado que >0 da valore >0.5 en probabilidad mapeada por la función logística.

Presentamos una distribución de los valores de os coeficientes para distintos valores de k:

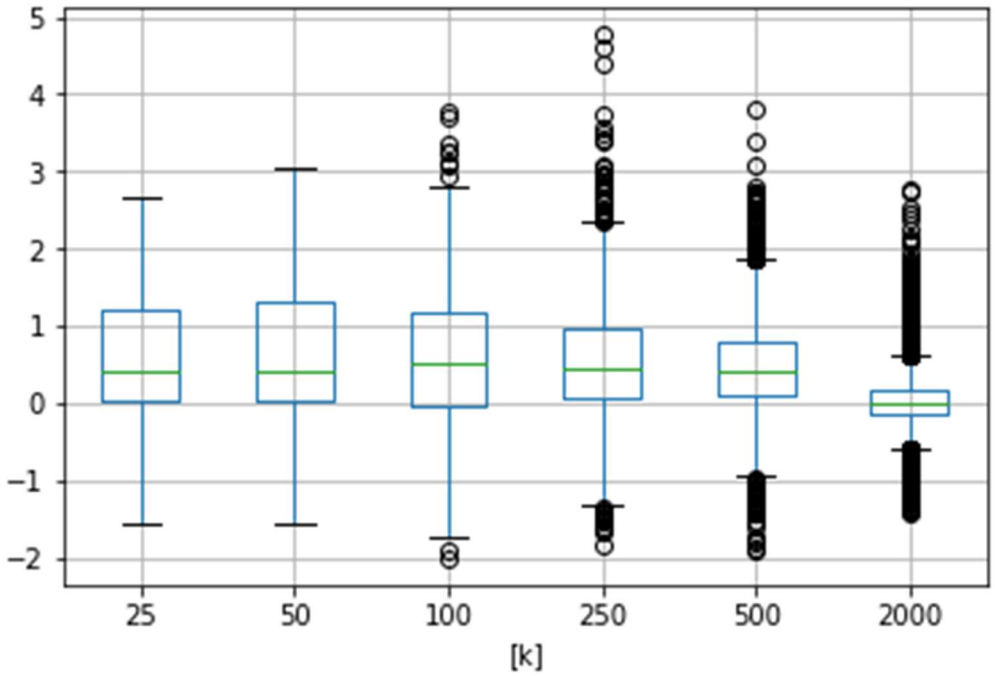

Ilustración 6:Boxplot valores coeficientes

Donde apreciamos que se son mayoritariamente positivos y por la normalización aplicada se encuentran entre valores de -2 y 5, aumentando sus valores según aumentamos el número de términos del conjunto (k).
Presentamos también los histogramas para cada k (tamaño del conjunto de atributos):

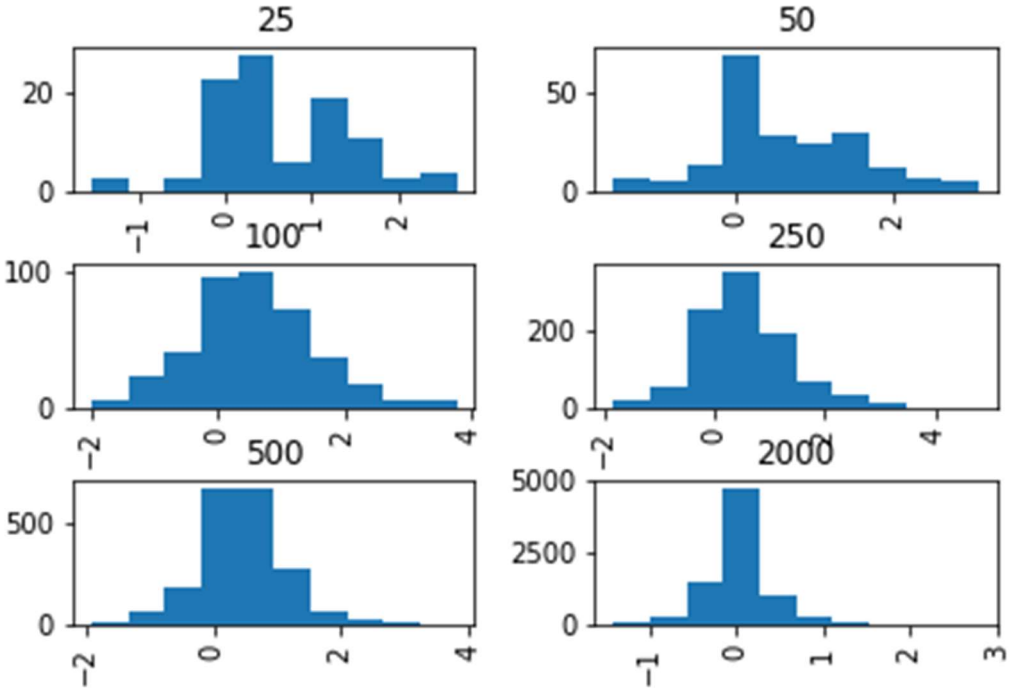

Ilustración 7:Histogramas coeficientes por número de atributos

5. Enriquecimiento del diccionario/Lexicon

5.1 Consideraciones iniciales

Tras la obtención de diferentes conjuntos de términos por diferentes métodos en el proceso anterior intentaremos generalizar/aumentar los mismos.

Los procedimientos recogidos en la bibliografía de referencia (1) para construir dichos lexicons a partir de un conjunto de palabras semillas/seed incluyen:

- Mediante el uso de vectorizaciones densas de palabras (tipo GLOOVE o word2vec) podemos:
  - Seleccionar las nuevas palabras por cercanía de coseno a cada una de las iniciales.
  - Podemos construir un eje como la diferencia entre el centroide de palabras positivas y el de las negativas y usar la proyección sobre ese eje como score para nuevas palabras.
  - Utilizar las distancias coseno y un subconjunto de palabras etiquetadas para construir un grafo de todas (pesos en aristas por cosenos) y aplicar algoritmo de labelling (positivo/negativo).

- Utilizar el corpus para detectar construcciones con conjunciones copulativas y adversativas en las que ya aparezca una palabra etiquetada, añadiendo en la misma categoría/peso las unidas por copulativas y en la inversa por adversativas.

- Utilizar un thesaurus tipo Wordnet [4] que permita localizar sinónimos y hipónimos de palabras.

- Construir nuevas palabras a partir de modificaciones morfológicas con sufijos y prefijos de la ya existentes.

Adicionalmente se puede partir de un lexicon o diccionario ya existente y utilizar las ponderaciones de las mismas en ciertas dimensiones (estos no incluyen un único peso, sino por diferentes sentimientos/afectos) para encontrar similitud a las palabras de origen.

Nosotros optaremos por la primera opción del uso de vectorizaciones densas (distancia de coseno) y el uso del thesaurus wordnet. Si dispusiéramos de otro conjunto de datos destino donde se fuera a aplicar el modelo podríamos investigar los términos del mismo y usar alguna de las otras técnicas recogidas.

5.2 Implementaciones previas

Para realizar facilitar la construcción de estos diccionarios extendidos hemos implementado una clase heredera del diccionario de Python que nos facilita la tarea teniendo en cuenta los posibles casos al añadir una nueva palabra:

[4] https://wordnet.princeton.edu/

- Si no existía, simplemente se añade.
- Si existía y su peso anterior era del mismo signo, pero menor en valor absoluto se actualiza.
- Si existía, pero su peso era de signo contrario, procedemos a eliminarla e incluirla dentro de lo que denominaremos listanegra. Dado que dicha palabra parece tener un comportamiento ambiguo y no deberá ser incluida en ningún caso.

En los casos que usemos un diccionario sin etiquetado POS (equivalente al preproceso LEM1) alimentaremos inicialmente nuestra lista negra con las stopwords (palabras auxiliares) para que no sean tenidos en cuenta.

5.3 Modo 1: word2vec y similarity cosine

Para cada palabra existente en nuestro diccionario de origen encontraremos encontraremos las n palabras (o hijos) mas similares y las añadiremos al diccionario con un peso igual al producto del peso de la palabra original (padre) multiplacado por la similaridad representada por el coseno (que posee valor 0…1).

Si no se repiten palabras o no entran en la lista negra nuestro diccionario final objetivo se vería enriquecido con un número de términos igual al número de hijos por palabra semillas/originales.

En este caso hemos inicializado la lista negra con los valores de las stopwords.

Presentamos los resultados para:

- Los conjuntos de términos obtenidos en la sección anterior mediante el preproceso LEM1 y selector mutual_info de 25, 50 y 100 términos.

- Obteniendo un numero de hijos de 10, 25 ,50 y 100 por palabra.

kOrigen	num_hijos	Nterm_NDict	Porcentaje sobre objetivo	Añadidos a BlackList	Porcentaje BlackList
25	10	188	75%	5	3%
25	25	379	61%	30	8%
25	50	637	51%	84	13%
25	100	1052	42%	188	18%
50	10	316	63%	25	8%
50	25	611	49%	99	16%
50	50	996	40%	218	22%
50	100	1590	32%	443	28%
100	10	542	54%	70	13%
100	25	1033	41%	193	19%
100	50	1619	32%	409	25%
100	100	2566	26%	785	31%

Tabla 6: Resultados ampliacion Lexion MODO 1

Donde apreciamos que según aumentamos el número de hijos y conjunto de palabras de origen disminuye el porcentaje alcanzado sobre el objetivo (términos reales añadidos respecto a objetivo). Esto se debe a que habrá términos que coincidirán en las diferentes búsquedas y además aumenta el porcentaje de palabras ambiguas incluidas en la black list.

5.4 Modo 2: Wordnet

Para este segundo método utilizaremos el thesausurs de Wordnet.

Obtendremos los symets relacionados con la palabra/lema de origen (que podrán ser múltiples) y recorreremos todos los lemas asociados a cada uno de ellos añadiéndolos al diccionario con el mismo peso que el original (se trata teóricamente de sinónimos).

Secundariamente habilitaremos la opción de añadir adicionalmente todos los hipónimos (especializaciones tipo perro->mastín) de cada uno de los Synets identificados así como aquellos relacionado mediante similar_tos. Como peso para los hipónimos supondremos el del padre reducido en un 10% y en el caso de los identificados mediante similar_tos por un 20%.

Dado que para este enriquecimiento usaremos los términos obtenidos mediante el preproceso LEM2 que incluye la categoría gramatical (con cuidado de la diferencias de nomenclatura entre nltk y wordnet), ofreceremos la opción de solo incluir palabras deducidas del mismo tipo (es decir si el origen era un adjetivo solo adjetivos).

Por último, incluimos también la opción de añadir los antónimos detectados que se realiza con un peso negado del peso original. Esto se restringirá a los adjetivos y adverbios que trataremos posteriormente.

Todos estos comportamientos serán parámetros del sistema de añadido y examinaremos los resultados como en el caso anterior para:

- Los conjuntos de términos obtenidos en la sección anterior mediante el preproceso LEM2 y selector mutual_info de 25, 50 y 100 términos.

- Combinación de parámetros del sistema: Incluir hyponimos, incluir obtenidas mediante similar_tos y restrigir solo a palabras de la misma categoría gramátical

kOrigen	bAddHypo	bAddOnlyOriginalPost	bAddSimilar	Nterm_NDict	Porcentaje BlackList
25	False	False	False	349	2%
25	False	False	True	382	2%
25	False	True	False	250	3%

25	False	True	True	250	3%
25	True	False	False	2108	2%
25	True	False	True	2141	2%
25	True	True	False	1782	2%
25	True	True	True	1782	2%
50	False	False	False	660	5%
50	False	False	True	732	5%
50	False	True	False	480	7%
50	False	True	True	483	7%
50	True	False	False	3221	5%
50	True	False	True	3293	5%
50	True	True	False	2725	5%
50	True	True	True	2728	5%
100	False	False	False	1234	4%
100	False	False	True	1440	3%
100	False	True	False	796	5%
100	False	True	True	840	4%
100	True	False	False	4655	6%
100	True	False	True	4861	6%
100	True	True	False	3626	6%
100	True	True	True	3670	6%

Tabla 7: Resultados ampliación Lexion MODO 2

Donde apreciamos lo siguiente:

- El uso de la relación adicional similar_tos resulta casi irrelevante, el número de términos obtenidos es el mismo prácticamente.
- El uso de los hyponimos de las palabras aumenta radicalmente el número de términos añadidos, multiplicándolo casi por 4.
- La obtención solo de términos del mismo tipo reduce el conjunto final en un 10/20% siendo más relevante cuando partimos de conjuntos pequeños (esto nos dice , como es lógico, que los sinónimos y hipónimos suelen ser de la misma categoría).

Adicionalmente a este proceso general hemos querido realizarlo sobre una restricción a adverbios y adjetivos, dado que normalmente estos son los que forman la base de los lexicons.
Restringimos nuestras semillas originales a aquellos que han sido calificados como estas categorías gramaticales y en este caso si añadimos antónimos y no cambiamos el parámetro similar_tos ni bAddHyppo al no ser relevante (no se incluyen para adjetivos y adverbios)
Usamos la lemanizacion LEM3 que solo utiliza dichos elementos:

kOrigen	Nterm_NDict	Porcentaje BlackList
25	165	0%
50	244	3%

100	367	1%

En este caso al no haber relación de hiponimia hace que el crecimiento del diccionario sea menor.

5.5 Evaluación de la bondad de los diccionarios obtenidos

El tamaño de los nuevos diccionarios nos puede servir como una medida del número de términos añadidos, pero intentaremos establecer alguna medida adicional para evaluar si dichos lexicons son útiles o no.
Evidentemente disponer de un segundo conjunto de datos etiquetados y medir la bondad del modelo sería lo óptimo; pero al no disponer del mismo optaremos por:

- Utilizar el diccionario de 2000 términos obtenido en los procesos anteriores como maestro y ver cuantos términos de los añadidos están en ese diccionario y coincide además el signo del peso.

- Aplicar sobre el conjunto de origen estos diccionarios y comparar los resultados con los obtenidos anteriormente.

5.5.1 Comparativa con maestro

Comprobamos de los términos añadidos cuantos de ellos ya estaban recogidos en el diccionario de 2000 términos antes obtenidos y de estos cuantos coinciden en el signo del peso.
Obtendremos:

- Numero y porcentaje de elementos añadidos que están incluidos en el maestro.
- Numero en maestro calificados con valor positivo (tendente a elevar el sentimiento negativo) y al revés en nuestro diccionario original.
- Numero en maestro calificados con valor negativo (tendente a reducir el sentimiento negativo) y al revés en nuestro diccionario original.
- Porcentaje de elementos clasificados erróneamente.

Para el MODO1 con word2vec y cosine:

kOrigen	num_hijos	Nterm_NDict	En maestro	%En maestro	POSNEG	NEGPOS	%NOCincidentes
25	10	188	104	55%	9	45	52%
25	25	379	174	46%	14	82	55%
25	50	637	265	42%	17	125	54%
25	100	1052	374	36%	21	186	55%
50	10	316	162	51%	16	70	53%
50	25	611	260	43%	22	111	51%
50	50	996	349	35%	34	152	53%
50	100	1590	444	28%	41	196	53%
100	10	542	228	42%	26	96	54%

100	25	1033	334	32%	34	137	51%
100	50	1619	403	25%	44	172	54%
100	100	2566	460	18%	36	225	57%

Para MODO2 mediante WordNet (solo ponemos con similar_tos=False por su irrelevancia):

kOrigen	Add Hypo	Only OriginalPost	Nterm_NDict	En maestro	%En maestro	POSNEG	NEGPOS	%NOCincidentes
25	False	False	349	93	27%	11	27	41%
25	False	True	250	75	30%	10	18	37%
25	True	False	2108	205	10%	23	81	51%
25	True	True	1782	173	10%	22	65	50%
50	False	False	660	163	25%	20	56	47%
50	False	True	480	129	27%	16	41	44%
50	True	False	3221	301	9%	33	122	51%
50	True	True	2725	248	9%	25	101	51%
100	False	False	1234	264	21%	42	80	46%
100	False	True	796	204	26%	34	59	46%
100	True	False	4655	386	8%	55	145	52%
100	True	True	3626	318	9%	43	121	52%

Para MODO2 restringido a adjetivos y adverbios (en este caso tomamos como maestro el total de términos):

kOrigen	Nterm_NDict	En maestro	%En maestro	POSNEG	NEGPOS	%NOCincidentes
25	165	36	21%	2	14	44%
50	244	46	18%	3	20	50%
100	367	61	16%	5	25	49%

Tabla 8: Comparativa diccionarios extendidos con maestro

Donde apreciamos que apreciamos conseguimos porcentajes entorno al 20% de palabras obtenidas que están en el maestro original, pero parece que no se correlaciona el signo del peso original con el deducido (en torno al 50%, aleatorio).

5.5.2 Prueba con datos originales

Dado que los el número de términos coincidentes entre el diccionarios existentes en el DataSet (lo que hemos denominado cruce con el maestro) y los diccionarios generados es muy pequeño, estimamos no son de aplicación estos diccionarios en este DataSet.
En caso de disponer de un segundo o de más casos y haber separado un conjunto de training y otro de test podríamos evaluarlos, pero no es el caso.

6.Nuevos Atributos

6.1 Consideraciones iniciales

A la hora de calificar el mensaje con un sentimiento negativo puede ser de gran utilidad el uso de otras características de la comunicación adicionales al propio contenido en palabras (o emotis).

Es de uso común utilizar otros atributos dependiendo del canal usado: uso de mayúsculas/signos de puntuación en texto escrito, volumen o tono en hablado…

Nótese que en nuestro caso el mensaje individual esta embedido en lo que denominaremos una conversación, por lo que también podremos usar medidas sobre esta para alimentar nuestro modelo.

Estos atributos pueden servir como entra al modelo inicial o usarse en un ensemble conjuntamente con el valor obtenido por nuestro clasificador original basado en lexicons.

6.2 Atributos propuestos

Como primera aproximación planteamos el uso de los siguientes atributos adicionales:

- Porcentaje de caracteres en mayúscula.
- Numero de apariciones de exclamaciones e interrogaciones repetidas. Contamos cuantas veces en el texto ocurren expresiones del tipo :???, !!! mediante expresiones regulares.
- Número de mensajes ya intercambiados en la misma conversación.
- Longitud total en caracteres de la conversación.

6.2 Evaluación

Para un primer análisis visual mostramos la distribución de los nuevos parámetros para el conjunto de mensajes positivos y negativos mediante un boxplot:

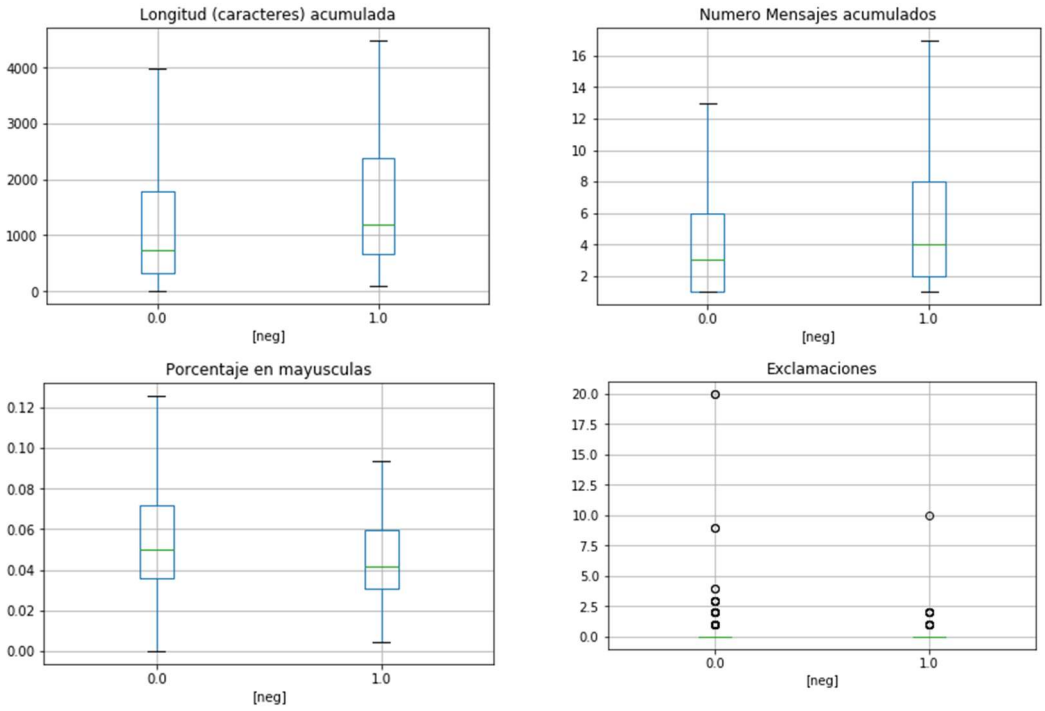

Ilustración 8: Boxplots distribución nuevos parámetros

Donde apreciamos que la los acumulados (en mensajes y texto) si parecen aumentar en aquellos calificados como negativos. Mientras que la media de caracteres en mayúscula y exclamaciones parece varía muy poco.

De todas formas, para evaluar en cuanto podría mejorar nuestro modelo realizaremos como en otros casos una regresión logística (balanceada por pesos a las categorías). Partiremos de una vectorización tdidf con la LEM1 antes propuesta y añadiremos a su entrada los nuevos valores calculados.

Mediremos Accurancy , Recall y precisión sobre el conjunto total.

	ORIGINAL	ENRIQUECIDO
Accurancy	0,957	0,960
Recall	0,386	0,401
Precision	0,976	0,976

Tabla 9 : comparativa medidas de bondad con nuevos parametros

Apreciamos una mejora cercana al 10% en el recall, si bien al no disponer de datos significativos para separar entrenamiento y test no sabemos si se debe a un overfitting más que a un comportamiento general deseado.

7. Conclusiones, Entregables y Herramientas

7.1 Conclusiones y otras consideraciones

La orientación de este TFM se basaba en intentar aplicar distintas técnicas de NLP y modelos de clasificadores supervisados para realizar un sentiment analysis de las comunicaciones de incidencias IT.
Se preveía como primera aproximación la construcción de un lexicon asociado que permitirá realizar dicha categorización, para posteriormente enriquecerlo con otros atributos (evaluación de saludos, uso de repeticiones/mayúsculas…) o modelos mas complejos (Deep learning, análisis sintactico…).

Como en la mayoría de casos de proyectos de ciencia de datos, el origen y calidad de los datos resulta un factor crítico en la obtención de resultados, si bien, debe entenderse que al tratarse de un ejercicio académico la finalidad es también el propio aprendizaje de los métodos usados.

Debemos por ello indicar que el DataSet de origen,

- por su reducido tamaño,
- muy pocos ejemplos que pudieran calificarse como de sentimiento negativo (y necesidad de etiquetarlo)
- y con muchos términos ofuscados que lo hacían poco legible,

creemos ha resultado un menoscabo en la obtención de resultados.

En cualquier caso, podemos indicar las siguientes conclusiones obtenidas de la investigación realizada:

- Selección de atributos. En el proceso aplicado para la selección de atributos hemos detectado lo siguiente:
    - Obtiene mejores resultados la técnica de mutual_info que el Chi2 evaluando la bondad a partir del recall de un logistic regresor. Los métodos no paramétricos suelen tener mejor comportamiento en estos ámbitos de NLP.

    - Al tener en cuenta todos los términos lemanizados (lo que hemos denominado como LEM1) obtiene mejor resultado que su separación por diferentes categorías gramaticales (LEM2); usando como métrica lo mismo que el caso anterior (recall logistic regressor) así como la media del conteo de términos para cada entrada.

- Asignación de pesos o sentido a los términos identificados como relevantes. Hemos partido de que el peso asignado en los coeficientes de un logistic regressor restringido al conjunto de atributos seleccionados se podía utilizar como peso asignado para el diccionario; pero la extensión de diccionarios parece contradecir dicha asignación.
  Entendemos que un peso individualizado respecto a un patrón sería más correcto, dado que el peso del que partimos se ve influido por todo el conjunto de términos escogido para realizar la regresión.

Debemos entender que en la construcción de lexicons y su extensión semisupervisada normalmente se parte de un conjunto y pesos realizado ad-hoc manualmente por expertos.

- Extensión de diccionarios:
    - Los procedimientos de extensión de diccionarios utilizados obtienen buenos resultados si consideramos la cardinalidad del conjunto de origen y de salida, obteniendo crecimientos incluso del 1000%.

    - La asignación de pesos a los nuevos términos generados no ha obtenido los resultados esperados, si bien su cardinalidad era pequeña (solo un 10% de los añadidos se reflejaban en el DataSet) y arrastra el problema de asignación de pesos antes comentado.

- El uso exclusivo de solo adverbios y adjetivos, como parece la práctica común en lexicons, no ha conducido a mejores resultados en este ejemplo, lo cual puede ser achacable a las restricciones del propio DataSet.

- El uso de atributos/medidas adicionales al conteo de palabras puede ser una entrada relevante para nuestro modelo. Estos pueden estar relacionados con el concepto global de conversación (medir numero de mensajes totales anteriores) o con el uso de ciertos tipos de caracteres (Exclamaciones, mayúsculas…).

Como posibles direcciones futuras para acometer el trabajo estimamos lo siguiente:

- En primer lugar, creo es requisito indispensable disponer de un mayor conjunto de datos y ejemplos etiquetados para abordar el proyecto.
  Esto permitiría el uso de otros estadísticos pará la selección de atributos así como usar el propio corpus para la extensión de los mismos, ya fuera mediante una vectorización densa del mismo u otras técnicas. EL uso de vectorizaciones o thesaurus generalistas para ámbitos de detección de sentimiento especifico no parece una solución correcta en un ámbito con su propio lenguaje característico.

  Este trabajo (14) realizado para microblogging y orientado al mundo de las finanzas puede resultar muy útil como guía de métodos a aplicar.
  En el , se parten de muchos mas datos y dos polaridades claras: positivas y negativas, no como en nuestro caso donde no teníamos un opuesto al cabreado o negativo, sino solo indiferente o negativo.

- En segundo lugar y por la casuística reflejada en las conversaciones, la aplicación de técnicas de chatbox (detección de utterances…) creemos puede ser de aplicación en este ámbito. Dado que podemos detectar en la mayoría de los casos que el enfado deriva en la afirmación de que se va a abandonar al proveedor o que se tomaran medidas legales.

7.2 Entregables

El TFM se basa en el presente documento, pero adicionalmente se incluirán los siguientes ficheros:

- Dataset (esta dentro de jupy para ser accesible a estos). Dentro de esta carpeta se incluirán los ficheros de datos utilizados en el trabajo, lo cual incluirá:
    - DataSet Original. Formado por cuatro ficheros csv: generic_malware.csv, iot_malware.csv, MDNS_vulnerabiltiy.csv y resolver_vulnerability.csv
    - Ficheros EXCEL auxiliares generados(volcado DataFrames):
        - Total.xlslx . Concatenación de los ficheros anteriores incluyendo columna de origen.
        - total_etiquetadoIMP.xlsx. Fichero anterior enriquecido con una columna (neg) donde se indican aquellos mensajes identificados -tras el proceso manual- como negativos.
        - valorsFeaturesSel.xlsx. Valores obtenidos para los conjuntos de selección de atributos.
        - valorsFeaturesCoef.xlsx. Valores de los coeficientes de regresión para cada termino escogido en las diferentes combinatorias.

- Auxiliares: Hojas Excel y otros elementos usados para el tratamiento de datos para su incorporación a este documento.

- Jupy. En esta carpeta se integran los notebook jupyter que se han utilizado como base para el desarrollo en Python y obtención de los datos y modelos que sirven de base a este TFM. Cada uno esta directamente relacionado con una sección de este documento:
    - analisisBase : análisis de DataSet aportado y primer modelo.
    - tok_vectorizador: Selección de términos/features.
    - ext_lexicon : extensión de los diccionarios.
    - nuevos_atributos: inclusión de atributos adicionales.

7.3 Herramientas

Se ha utilizado Python como lenguaje de programación principal usando las siguientes librerias de uso común para el NLP y ciencia de datos:

- Nltk
- Gensim
- Sklearn
- Pandas y numpy
- Matplotlib

Los desarrollos realizados se presentan como notebooks jupyter , integrando el propio código con comentarios y gráficas asociadas.

Referencias

1. **Martin, Dan Jurafsky and James H.** *Speech and Language Processing (3rd ed. draft).* 2018.
2. *Sentiment Analysis:A Comparative Study On Different Approaches.* **Devika M D[a]*, Sunitha C[a], Amal Ganesh.** Fourth International Conference on Recent Trends in Computer Science & Engineering.
3. *Affective Computing.* **Picard, R. W.** 1995.
4. *A Comparative Study of Sentiment Analysis Techniques.* **Pooja Dinkar Shinde, Dr. Sunil Rathod.** International Journal of Innovations & Advancement in Computer Science.
5. *Enhancing deep learning sentiment analysis with ensemble techniques in social applications.* **Oscar Araque ∗, Ignacio Corcuera-Platas, J.Fernando Sánchez-Rada, CarlosA. Iglesias.** Vols. Expert Systems With Applications 77 (2017) 236–246.
6. *A Comparative Study on Twitter Sentiment Analysis: Which Features are Good?.* **Koto, Fajri & Adriani, Mirna.** 2015. Conference: Springer in The 20th International Conference on Applications of Natural Language To Information Systems (NLDB 2015),, At Passau, Germany, Volume: 453-457.
7. *A Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation.* **(lionel.brunie@insa-lyon.fr)∗, Ana¨ıs Collomb (anais.collomb@insa-lyon.fr)∗ Crina Costea (crina.costea@insa-lyon.fr)∗ Damien Joyeux (damien.joyeux@insa-lyon.fr)∗ Omar Hasan (omar.hasan@insa-lyon.fr)∗ Lionel Brunie.**
8. **Antonio Gulli, Sujit Pal.** *Deep Learning with Keras.* 2016.
9. *Convolutional Neural Networks for Sentence Classification.* **Kim, Yoon.** 2014.
10. *Convolutional Neural Networks for Sentiment.* **Salinca, Andreea.**
11. **Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang,.** *Recursive Deep Models for Semantic Compositionality.*
12. **Quoc Le, Thomas,Tomas Mikolov.** *Distributed Representations of Sentences and Documents.*
13. *Sentiment Analysis in Tickets for IT Support.* **Cássio Castaldi Araujo Blaz, Karin Becker.** 2016. IEEE/ACM 13th Working Conference on Mining Software Repositories.
14. **Nuno Oliveiraa, , Paulo Corteza, Nelson Areal.** *Stock market sentiment lexicon acquisition using microblogging data and statistical measures.* s.l. : ALGORITMI Centre, Department of Information Systems, University of Minho,.