UNIVERSIDAD DE GRANADA

MÁSTER DE CIENCIA DE DATOS E INGENIERÍA DE COMPUTADORES

CURSO ACADÉMICO 2019-2020

MINERÍA DE MEDIOS SOCIALES

Minería de opinión con KNIME.

Aplicación de algoritmos de Minería de opiniones y sentimientos con KNIME.

Nicolás Cubero

15 de Mayo de 2020

Índice

1. Introducción

En este proyecto, se tratará la aplicación de diversos métodos y algoritmos de Minería de opinión sobre una colección de comentarios recogidos del sitio web *IMDb* para los cuales, se recoge también un etiquetado sobre la polaridad de cada opinión (positiva o negativa).

Se hará uso de diferentes métodos para evaluar la polaridad cada opinión recogida en el conjunto de datos y se evaluará la eficacia de cada método mediante su comparación con la polaridad original etiquetada.

Primeramente se hacen uso de los lexicones *MPQA Subjectivity Lexicon* [1], *SentiWordNet 3.0* [2] y *SenticNet 5.0* [3] para etiquetar la polaridad de cada término en cada opinión y con ello, determinar la polaridad global de cada opinión y evaluar la eficiencia de este etiquetado respecto de la polaridad original.

Por último, se compararán los rendimientos de estos métodos con los obtenidos por modelos de clasificación basados en *Support Vector Machine* (SVM) y redes neuronales *Multilayer Perceptron* (MLP) que se cosntruirán reservando el 70 % de los documentos para el entrenamiento de los modelos y el 30 % restante para su evaluación.

2. Preprocesamiento de lexicones

En primer lugar, se desea tratar y explicar el preprocesamiento y formateo de los lexicones *SentiWordNet 3.0* y *SenticNet 5.0* que se lleva a caso como paso previo a su utilización.

2.1. Preprocesamiento del lexicon *SentiWordNet 3.0*

De este lexicon resulta únicamente relevante los términos recogidos en la columna `SynsetTerms` y las puntuaciones de polaridad positiva y negativa asociados a los mismos y recogidos, de forma respectiva, en las columnas `PosScore` y `NegScore`.

[1] Enlace al sitio web sobre el lexicon MPQA Subjectivity Lexicon: http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

[2] Enlace al repositorio GitHub con el lexicon SentiWordNet 3.0: https://raw.githubusercontent.com/aesuli/SentiWordNet/master/data/SentiWordNet_3.0.0.txt

[3] Enlace al sitio web los lexicon de SenticNet: https://www.sentic.net/downloads/

Primeramente, para cada término se determina una etiqueta única que refleje su polaridad positiva o negativa en base a estas puntuaciones de polaridad positiva y negativa. Se hace uso de un nodo de ejecución Java (*Java Snippet*), para determinar la polaridad de acuerdo a la siguiente regla:

$$polarity = \begin{cases} \text{“}positive\text{”} & SiPosCore > NegScore \\ \text{“}negative\text{”} & SiPosCore < NegScore \\ \text{“}neutral\text{”} & SiPosCore = NegScore \end{cases} \quad (1)$$

Esta polaridad se coloca en una nueva columna `polarity`. Una vez calculada, se filtran las columnas `SynsetTerms` y `polarity` del conjunto de datos con el nodo *Column Filter*.

A continuación, dado que cada palabra en los términos de la columna `SynsetTerms` lleva concatenada una etiqueta de rango, se hacen uso de expresiones regulares para eliminar dichas etiquetas de rango y obtener únicamente las palabras de cada término. Para este fin, se configura otro nodo *Java Snippet* para realizar dicha transformación con expresiones regulares.

Por último, se separan los términos positivos de los negativos y se devuelve cada uno por una salida distinta del metanodo de preprocesamiento del lexicon, haciendo uso de nodos *Column Filter* para filtrar ambos tipos de términos en base a sus valores de la columna `polarity` que se calculó.

2.2. Preprocesamiento del lexicon *SenticNet 5.0*

Este lexicon recoge, para cada término, una etiqueta de polaridad con los valores positivo y negativo (*positive* y *negative*) junto a la intensidad asociada a dicha polaridad. De este conjunto de datos interesa únicamente los términos y la polaridad almacenados, respectivamente, en las columnas *concept* y *polarity*, por lo que se hace uso del nodo *Column Filter* para filtrarlos.

Por su parte, los términos recogidos en la columna *concept* hacen uso de “_” para reflejar el espacio entre las palabras que conforman el propio término en lugar del propio espacio, por lo que es necesario reemplazar los “_” por espacios, para lo que se hace uso de un nodo *Java Snippet* para reelizar este reemplazamiento.

Por último, al igual que con el anterior lexicon se separan los términos positivos y los negativos del conjunto original y se devuelven en dos salidas di-

ferentes del nodo de preprocesamiento del lexicon, haciendo uso nuevamente de nodos *Column Filter* y filtrando por los valores de la columna *polarity*.

3. Evaluación y comparación de rendimientos

Se evalúan los rendimientos de cada método usado para la predición de la polaridad de las opiniones comparandolas con el etiquetado original recogido en el *dataset* de *IMDb*.

3.1. Rendimiento con lexicon *MPQA Subjectivity Lexicon*

La matriz de confusión obtenida en la clasificación mediante el lexicon *MPQA Subjectivity Lexicon* es la siguiente:

		Opinión Real	
		+	-
Opinión Predicha	+	815	447
	-	119	470

Tabla 1: Matriz de confusión del lexicon MPQA Subjectivity Lexicon

Por su parte, las métricas con este lexicon resultan las siguientes:

	Opinión	
	+	-
Precisión	0.646	0.798
Recall	0.873	0.513
Specificity	0.513	0.873
F-meausure	0.742	0.624
Accuracy	0.694	

Tabla 2: Métricas de rendimiento obtenidas con el lexicon MPQA Subjectivity Lexicon

Las métricas nos revelan un rendimiento medio por parte de este método que presenta una cierta mayor dificultad para identificar opiniones con polaridad negativa tal y como se refleja en la matriz de confusión y en la métrica *F-meausure*.

3.2. Rendimiento con lexicon *SentiWordNet 3.0*

La matriz de confusión obtenida en la clasificación mediante el lexicon *SentiWordNet 3.0* es la siguiente:

		Opinión Real	
		+	-
Opinión Predicha	+	672	477
	-	279	480

Tabla 3: Matriz de confusión del lexicon SentiWordNet 3.0

Por su parte, las métricas con este lexicon resultan las siguientes:

	Opinión	
	+	-
Precisión	0.585	0.632
Recall	0.707	0.502
Specificity	0.502	0.707
F-meausure	0.64	0.559
Accuracy	0.604	

Tabla 4: Métricas de rendimiento obtenidas con el lexicon SentiWordNet 3.0

Las métricas con este lexicon muestran un rendimiento algo más reducido que con el lexicon *MPQA Subjectivity Lexicon.*

3.3. Rendimiento con lexicon *SenticNet 5.0*

La matriz de confusión obtenida en la clasificación mediante el lexicon *SenticNet 5.0* es la siguiente:

		Opinión Real	
		+	-
Opinión Predicha	+	998	994
	-	2	6

Tabla 5: Matriz de confusión del lexicon SenticNet 5.0

Por su parte, las métricas con este lexicon resultan las siguientes:

	Opinión	
	+	-
Precisión	0.501	0.75
Recall	0.998	0.006
Specificity	0.006	0.998
F-meausure	0.667	0.012
Accuracy	0.502	

Tabla 6: Métricas de rendimiento obtenidas con el lexicon SenticNet 5.0

Las métricas con este lexicon muestran una reducida capacidad predictiva hacia la polaridad negativa, es decir, el método no es capaz de identificar documentos de polaridad negativa con este lexicon, por lo que el rendimiento es todavía inferior que con los anteriores lexicones.

3.4. Rendimiento con modelo *Support Vector Machine*

La matriz de confusión obtenida en la clasificación mediante un modelo SVM configurado con $C = 0{,}9$ y kernel *RBF* con $sigma = 0{,}2$ es la siguiente:

		Opinión Real	
		+	-
Opinión Predicha	+	250	27
	-	50	273

Tabla 7: Matriz de confusión del modelo SVM de kernel RBF con C=0.9 y sigma=0.2

Las métricas de este modelo se muestran en la siguiente tabla:

	Opinión	
	+	-
Precisión	0.903	0.845
Recall	0.833	0.91
Specificity	0.91	0.833
F-meausure	0.887	0.876
Accuracy	0.872	

Tabla 8: Métricas de rendimiento obtenidas con el modelo SVM de kernel RBF con C=0.9 y sigma=0.2

El rendimiento ofrecido por este modelo sobre el conjunto de test es muy superior que los métodos anteriores basados en lexicones y alcanza un valor elevado.

3.5. Rendimiento con modelo *Multilayer Perceptron*

La matriz de confusión obtenida en la clasificación mediante un modelo MLP configurado con 1 capa oculta y 56 nodos en esta capa oculta es la siguiente:

		Opinión Real	
		+	-
Opinión Predicha	+	259	68
	-	41	232

Tabla 9: Matriz de confusión del modelo MLP con una capa oculta de 56 nodos.

Las métricas de este modelo se muestran en la siguiente tabla:

	Opinión	
	+	-
Precisión	0.792	0.85
Recall	0.863	0.773
Specificity	0.773	0.863
F-meausure	0.826	0.81
Accuracy	0.818	

Tabla 10: Métricas de rendimiento obtenidas con el modelo MLP con una capa oculta de 56 nodos.

El rendimiento ofrecido por este modelo también es notoriamente superior a los rendimientos obtenidos con los lexicones, aunque es algo inferior al rendimiento del modelo SVM. Se aprecia por tanto que el modelo MLP presenta una mayor dificultad para ajustarse a los datos que el modelo SVM.

Concluímos como resultado de estos experimentos, que los modelos de clasificación, tanto SVM como MLP, ofrecen buenos resultados en comparación a los métodos basados en lexicones.

Ambos métodos de clasificación, al alimentarse con la representación vectorial de los documentos con las frecuencias TF-IDF de cada término, permiten

explotar más la información sobre la aparición y relevancia de los términos en cada documento además de la relación entre los mismos, mientras que los lexicones, al basarse en las polaridades estimadas para cada término, asignadas por promedio en el lexicon según el análisis del uso del idioma, con lo que no se tiene en cuenta su significado concreto dado por el contexto del documento, introducen un mayor error en la clasificación.