UNIVERSIDAD DE GRANADA

MÁSTER DE CIENCIA DE DATOS E INGENIERÍA DE COMPUTADORES

CURSO ACADÉMICO 2019-2020

MINERÍA DE MEDIOS SOCIALES

Minería de textos con KNIME.

Aplicación de algoritmos de Minería de Textos con KNIME sobre un conjunto de tweets.

Nicolás Cubero

15 de Mayo de 2020

Índice

1. Introducción

En este proyecto se tratará la aplicación de un conjunto de algoritmos de **Minería de textos** con el software *KNIME* sobre un conjunto de *tweets* que recogen distintas opiniones sobre la actuación del Gobierno español y las autoridades sanitarias ante la crisis sanitaria provocada por el virus *Sars-CoV-2* o *COVID-19*, denominado de forma común como *coronavirus*.

Los *tweets* son recopilados con ayuda de la herramienta de *scraping Twitter Intelligence Tool* (TWINT) [1] y exportados a un fichero *csv*, desde el que se importa cómodamente a *KNIME*.

Sobre este conjunto de *tweets* se tratará diversas técnicas de preprocesamiento de textos (*tokenización*, *Part of Speech*, lemantización, etc) para limpiar, enriquecer los textos y prepararlos para su procesamiento por distintas técnicas de Minería de textos.

Una vez preprocesado el texto, se aplicará, en primer lugar, un proceso de extracción de los términos más frecuentes y su visualización con gráficos *Tag Cloud* y, por último, la aplicación de **Minería de Reglas de Asociación** para tratar de descubrir conjuntos de términos que tienden a aparecer juntos en los *tweets*.

2. Documentos tratados

Como se ha mencionado anteriormente, se tratarán todos los *tweets* enviados durante el mes de Abril de 2020 (desde el 1 de Abril hasta el 1 de Mayo) en el territorio español (todos los *tweets* registrados en una localización geográfica comprendida en un radio menor o igual a 600 km desde Madrid) escritos en lenguaje catellano y que tratan diversas opiniones o comentarios sobre la actuación del Gobierno español y las autoridades sanitarias ante la crisis sanitaria provocada por el *coronavirus*.

Para ello, se recopilan los *tweets* que incluyan alguno de los términos del siguiente conjunto: “gobierno”, “estado”, “españa”, “gestion”, “autoridad” y “autoridades”; de forma simultánea con alguno de los siguientes términos: “coronavirus”, “virus”, “covid”, “covid19”, “pandemia”, “crisis sanitaria”.

[1] Enlace al repositorio GitHub de TWINT: https://github.com/twintproject/twint

Se recopilan un total de 7240 *tweets* con la ayuda de la herramienta *TWINT* y se exportan a un fichero *csv*.

3. Algoritmos aplicados

Tras la lectura de los documentos con los *tweets* desde el fichero *csv*, se procede primeramente a realizar un **preprocesamiento general** cuya salida es utilizada como entrada tanto en la tarea de obtención de los términos más frecuentes como en la tarea de minería de reglas asociativas:

- **Preprocesamiento**: Se realiza un preprocesamiento de los textos de los *tweets* que se enfoca de forma global para ambas tareas de minería de textos:

    1. **Carga de *tweets* y converión de cada *tweet* a documento**: El fichero *csv* contiene los *tweets* extraídos, junto a su identificador, identificador y *username* del usuario que lo envío, además del lugar registrado donde se envió el *tweet*.

       En este primer paso se pretende seleccionar de este fichero cargado únicamente los *tweets* y convertir cada uno de ellos en un documento para su tratamiento por el resto de algoritmos.

       Estas operaciones se realizan con los nodos **string to document** para transformar los *tweets* en documentos y *Column Filter* para seleccionar la columna con los documentos.
    2. ***Part of Speech Tagging* y tokenización**: Con el nodo **POS Tager** y haciendo uso del *tokenizador* de *Stanford* para lenguaje castellano (*StanfordNLP Spanish Tokenizer*), se realiza el *Part of Speech Tagging* de cada documento/*tweet* y se divide en *tókens*.
    3. **Conversión a minúscula**: Los textos de los documentos se pasan a minúscula con el nodo **Case converter**.
    4. **Eliminación de símbolos de puntuación**: Se eliminan todos los signos de puntuación (puntos, comas, signos de interrogación y de exclamación, etc) que contengan los textos con el nodo **Punctuation Erasure**.
    5. **Eliminación de palabras de parada**: Se eliminan las palabras carentes de significado útil de los documentos/ *tweets* con el nodo **Stop Word Filter**.

6. **Eliminación de palabras muy poco frecuentes** que tengan una frecuencia inferior a 5 con el nodo **N chars filter**.
7. **Filtrado de sustantivos**: Dado que para la obtención de los términos más frecuentes y para la búsqueda de términos asociados sólo se consideran relevantes los nombres y sustantivos, se eliminan de los documentos todas las palabras no calificadas en el *Part of Speech* como sustantivo con el nodo **Tag Filter**.
8. **Lematización**: Se aplica lematización para devolver cada palabra de los *tweets* a su forma canónica con la utilidad **Stanford Lemmatizer**.

De este modo, como resultado de todas estas operaciones de preprocesamiento, se obtienen un conjunto de documentos por cada *tweet* con los términos nominales en su forma canónica para su utilización por los algoritmos de minería de textos que se aplican a continuación.

- **Búsqueda de términos más frecuentes**: Esta tarea pretende determinar los términos más usados en los *tweets* y visualizarlos con gráficos *Tag cloud*.

  1. **Generación de la bolsa de palabras**: Con los términos de los documentos de cada *tweet* se elabora la bolsa de palabras con ayuda del nodo **Bag of Words Creator**.
  2. **Cálculo del vector de frecuencias TF-IDF de cada término**: Para cada palabra de cada documento se determina su frecuencia TF-IDF mediante los nodos **TF** para el Cálculo de la frecuencia relativa de cada término en su documento, el nodo **IDF** para el cálculo de la inversa de la frecuencia de cada término y un nodo de operación (**Java Snnipet**) para obtener el producto *TF* e *IDF*.
  3. **Selección de los términos más frecuentes y visualización en un *Tag Cloud***: Se seleccionan los 30 términos más frecuentes según la frecuencia *TF-IDF* con el nodo *Frequency Filter* y elaboración de un gráfico *Tag Cloud* en imagen *png*.

- **Extracción de términos asociados**: Finalmente, se tratará de encontrar conjunto de términos nominales en los *tweets* que tienden a aparecer conjuntamente, para lo cual, se aplicará **búsqueda de reglas asociativas** más relevantes sobre los términos de cada documento.

  Estas reglas asociativas son almacenadas en un fichero *csv* junto a su puntuación de soporte, confianza y *lift*.

1. **Generación de transaciones**: Se preprocesan los documentos de términos nominales para transformarlos en transacciones con los términos nominales como ítems. Se hace uso del nodo **Cell Splitter** para convertir cada documento en un agregado de términos.
2. **Búsqueda de reglas asociativas**: Se aplica minería de búsquedas asociativas con el método *A priori* considerando un soporte mínimo de 0.001 y una confianza mínima de 0.99 mediante el nodo **Association Rule Learner**.
3. **Formateo de salida y salvado en fichero *csv***: El nodo **Association Rule Learner** devuelve una tabla con la puntuación de soporte, confianza y *lift* de cada regla junto con el antecedente y consecuente de la regla en columnas separadas que aparecen ordenadas en un orden no comveniente para el almacenamiento de las reglas y su visualización, por lo que se aplican una serie de nodos para combinar en una sóla columna el antecedente y consecuente de la regla separados con `-->` (nodo **Column Combiner**) y colocar esta columna antes que las columnas con las puntuaciones de cada regla (nodo **Column Resorter**).

   Para acabar, la tabla con las reglas y su puntuación se alamcenan en un fichero *csv* con el nodo **CSV Writer**.

La búsqueda de reglas de asociación permitió encontrar 3314 reglas con relaciones de términos con un soporte comprendido entre 0.0011051250172675784 y 0.005249343832020997 y una confianza de 1 en la mayoría de ellas.