Breve estudio de la clasificación de artículos periodísticos deportivos con word2vec

Antonio Murillo Sevillano
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
antmursev@alum.us.es

Ruben Romero Sanchez
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
rubromsan@alum.us.es

En este breve ensayo se tratará la clasificación de fragmentos de texto basado en técnicas de aprendizaje supervisado y no supervisado. Usaremos como ejemplificación un conjunto de artículos deportivos de la BBC, clasificados según deportes.

Se verán técnicas como la tokenización de los textos, stemming sobre el corpus utilizando la librería NLTK, el modelo word2vec entre otros, con el fin de utilizar técnicas de clustering para agrupar los documentos y discutir si los grupos creados guardan relación con las etiquetas originales.

Palabras Clave— Corpus, tokenizar, stemming, cluster.

I. Introducción

Las técnicas de procesamiento de lenguaje natural son un tema candente en la sociedad actual. Están muy presente en toda la tecnología que nos rodea y cada vez son más usados y eficientes. Los motores de búsqueda como Google, hasta la recomendación de productos o la traducción automática son solo algunos ejemplos de cómo Word2Vec y las técnicas de procesamiento de lenguaje natural en general están presentes en elementos cotidianos.

El objetivo del estudio es la creación de un sistema capaz de etiquetar textos en función de su contenido utilizando técnicas de aprendizaje supervisado y no supervisado.

Los datos a tratar constan de 737 documentos de la página web BBC Sport, correspondientes a noticias de deportes de cinco áreas temáticas de 2004 a 2005. Las clases o etiquetas son 5 (athletics, cricket, football, rugby, tennis). Esto constituirá el corpus.

Aplicaremos diversas técnicas sobre el conjunto de datos, usando diversas bibliotecas con el fin de transformar los textos en vectores de palabras que porten información relevante para el problema.

II. Preliminares

A. Métodos empleados

- Tokenización: suele ser el primer paso en muchas tareas de procesamiento de lenguaje natural. Es el proceso de dividir un texto en unidades más pequeñas llamadas tokens. Estos tokens pueden ser palabras individuales, frases, caracteres o cualquier otra unidad que se considere relevante para el análisis de texto. El resultado de aplicar la técnica es una secuencia de tokens que se utiliza para análisis posteriores. En este proyecto en el proceso de tokenización se eliminarán también las stop words (o palabras que no portan información sobre el problema). Por ejemplo, en la clasificación de documentos se espera una mayor capacidad predictora de la clase para la palabra “tigre” que para la palabra “pero”.
- Stemming: proceso en el procesamiento de lenguaje natural que implica reducir las palabras a su forma raíz o base. El objetivo del stemming es reducir las palabras a su forma principal, lo que ayuda a eliminar las variaciones morfológicas y permite tratar las palabras con la misma raíz como si fueran idénticas. El proceso busca eliminar los sufijos y prefijos de las palabras para obtener la raíz. Por ejemplo, el stemming podría convertir las palabras "corriendo", "correrá" y "corrió" en la raíz "correr". Esto facilita el análisis del texto, ya que se considera que todas las palabras con la misma raíz tienen un significado similar.
- Word2vec: Word2Vec es un modelo popular de procesamiento de lenguaje natural que se utiliza para aprender representaciones vectoriales de palabras. El objetivo principal de Word2Vec es capturar relaciones semánticas y sintácticas entre las palabras en un corpus de texto. El modelo se basa en la idea de que las palabras que aparecen en contextos similares tienden a tener significados similares. Existen dos variantes de Word2vec, ambas utilizan una red neuronal de 3 capas (1 capa de entrada, 1 capa oculta, 1 capa de salida): Common Bag Of Words (CBOW) y Skip-gram. Como su nombre indica, el método se basa en transformar palabra en vectores donde las palabras con significados contextos similares tendrán vectores cercanos en el espacio.
- Clustering: también conocido como análisis de grupos o agrupamiento, es una técnica de aprendizaje automático y minería de datos que se utiliza para identificar patrones y agrupar objetos similares en conjuntos más grandes. El objetivo principal del clustering es dividir un conjunto de datos en grupos o clusters, de manera que los objetos dentro de un mismo grupo sean más similares entre sí que con los objetos de otros grupos.

III. METODOLOGÍA

A continuación, un ejemplo de uso de listas numeradas:

1) Constitución del corpus: el conjunto de datos de origen tuvo que ser modificado y adaptado para su tratamiento, ya que venia en un formato raw. Procesamos los datos para su fácil tratamiento convirtiendo las 5 carpetas con las 5 clasificaciones que contenían los artículos en formato .txt en un solo .csv, un archivo mucho mas simple de usar y procesar.

2) Tokenización y eliminación de stopwords: tras haber procesado el corpus creamos una función que tokenize los textos. Esto consiste en sustituir todo lo que no sean letras por espacios, pasar las palabras a minúsculas y dividirlas en individuales, eliminar caracteres sueltos y por ultimo eliminar las palabras vacías o stop words de una lista de stop words descargadas de la librería NLTK.

3) Stemming: tras haber creado la función de tokenización de textos procedemos a crear una función de stemming, en la que usando la salida de la función anterior(tokenización) y con la ayuda del algoritmo de stemming snowball procedemos a reducir las palabras a su raíz o forma base. Es importante tener en cuenta que el stemming no siempre produce una raíz válida o legible en sí misma. En cambio, se enfoca en reducir las palabras a una forma común para el análisis.

4) Creacion del vocabulario final: Establecemos un umbral para que solo nos apararezcan las palabras que más se repiten, así nos centramos en las palabras más importante o informativas.

5) Word2Vec: con ayuda de la librería gensim creamos el modelo Word2Vec, cuyo objetivo es detectar palabras sinónimas o sugerir palabras adicionales para una frase sin terminar. Configuramos el modelo con los parametros descritos en el codigo.

6)Clustering:

Una vez obtenidas las representaciones vectoriales de las palabras, se ha utilizado el algoritmo de clustering KMeans para agrupar los documentos del conjunto de datos en clusters. KMeans es un algoritmo de clustering particional que divide el conjunto de datos en k clusters, donde k es un parámetro especificado por el usuario. El algoritmo asigna cada documento a uno de los k clusters minimizando la suma de las distancias al cuadrado entre cada documento y el centroide del cluster al que pertenece.

Para visualizar los resultados del clustering, se ha utilizado la técnica de reducción de dimensionalidad t-SNE para proyectar los vectores de características de alta dimensionalidad en un espacio bidimensional. t-SNE es una técnica no lineal que preserva las relaciones entre los puntos en el espacio original y permite visualizar la estructura global y local del conjunto de datos en un gráfico bidimensional. Los resultados del clustering se han visualizado en un gráfico de dispersión, donde cada punto representa un documento y los colores indican la pertenencia a uno u otro cluster.

Además del algoritmo KMeans, también se ha utilizado el algoritmo de clustering jerárquico aglomerativo para agrupar los documentos. Este algoritmo construye una jerarquía de clusters fusionando iterativamente los pares de clusters más cercanos hasta que todos los documentos pertenecen a un único cluster. El resultado del clustering jerárquico aglomerativo se puede visualizar utilizando un dendrograma, que muestra la estructura jerárquica de los clusters y permite cortar el dendrograma en diferentes niveles para obtener diferentes soluciones de clustering.

En cuanto a la evaluación del modelo, se han propuesto diferentes métricas para comparar las etiquetas asignadas por el modelo con las etiquetas originales del conjunto de datos. Estas métricas incluyen el índice Rand ajustado y la información mutua ajustada, que miden la similitud entre dos asignaciones de etiquetas y permiten determinar si los grupos creados por el modelo guardan relación con las etiquetas originales.

IV. RESULTADOS

En esta sección se detallará tanto los experimentos realizados como los resultados conseguidos:

Para este experimento se han utilizado un total de 737 documentos (noticias deportivas) divididos en 5 categorías diferentes. Los resultados finales que se esperan son, utilizando clustering, una representación gráfica de los documentos agrupados por grupo y ver si guardan relación. Antes de pasar a la parte final del clustering, se realizan otras operaciones que también arrojan resultados interesantes y que además tendremos que usar en el futuro.

1.- Construcción del corpus: Nos arroja un csv dividido por categorías y con el texto de cada noticia:

2.- Tokenización del corpus y eliminación de stopwords: Se observa un texto antes y después del proceso de tokenización:

```
Artículo sin tokenizar: O'Sullivan could run in Worlds. Sonia O'Sullivan has indicated that she would like to participate in ne
xt month's World Cross Country Championships in St Etienne. Athletics Ireland have hinted that the 35-year-old Cobh runner may
be included in the official line-up for the event in France on 19-20 March. Provincial teams were selected after last Saturday'
s Nationals in Santry and will be officially announced this week. O'Sullivan is at present preparing for the London marathon on
17 April. The participation of O'Sullivan, currently training at her base in Australia, would boost the Ireland team who won t
he bronze three years agio. The first three in last Saturday, Jolene Byrne, Maria McCambridge and Fionnualia Britton, ar
e automatic selections and will most likely form part of the long-course team. O'Sullivan will also take part in the Bupa Great
Ireland Run on 9 April in Dublin.

Artículo tokenizado: ['sullivan', 'could', 'run', 'worlds', 'sonia', 'sullivan', 'indicated', 'would', 'like', 'participate', '
next', 'month', 'world', 'cross', 'country', 'championships', 'st', 'etienne', 'athletics', 'ireland', 'hinted', 'year', 'old',
'cobh', 'runner', 'may', 'included', 'official', 'line', 'event', 'france', 'march', 'provincial', 'teams', 'selected', 'last',
'saturday', 'nationals', 'santry', 'officially', 'announced', 'week', 'sullivan', 'present', 'preparing', 'london', 'marathon',
'april', 'participation', 'sullivan', 'currentily', 'training', 'base', 'australia', 'would', 'boost', 'ireland', 'team', 'bron
ze', 'three', 'years', 'agio', 'first', 'three', 'santry', 'last', 'saturday', 'jolene', 'byrne', 'maria', 'mccambridge', 'fion
nualia', 'britton', 'automatic', 'selections', 'likely', 'form', 'part', 'long', 'course', 'team', 'sullivan', 'also', 'take',
'part', 'bupa', 'great', 'ireland', 'run', 'april', 'dublin']
```

3.- Stemming: es un método para reducir una palabra a su raíz. Aquí un ejemplo de stemming con el árticulo anterior:

```
['sullivan', 'could', 'run', 'world', 'sonia', 'sullivan', 'indic', 'would', 'like', 'particip', 'next', 'month', 'world', 'cro
ss', 'countri', 'championship', 'st', 'etienn', 'athlet', 'ireland', 'hint', 'year', 'old', 'cobh', 'runner', 'may', 'includ',
'offici', 'line', 'event', 'franc', 'march', 'provinci', 'team', 'select', 'last', 'saturday', 'nation', 'santri', 'offici', 'a
nnounc', 'week', 'sullivan', 'present', 'prepar', 'london', 'marathon', 'april', 'particip', 'sullivan', 'currentili', 'train',
'base', 'australia', 'would', 'boost', 'ireland', 'team', 'bronz', 'three', 'year', 'agio', 'first', 'three', 'santri', 'last',
'saturday', 'jolen', 'byrn', 'maria', 'mccambridg', 'fionnualla', 'britton', 'automat', 'select', 'like', 'form', 'part', 'long
', 'cours', 'team', 'sullivan', 'also', 'take', 'part', 'bupa', 'great', 'ireland', 'run', 'april', 'dublin']
```

4.- Word2Vec: Tras aplicar Word2Vec hemos ejecutado algunas funciones para ver ejemplos y su correcto funcionamiento:

-Palabras similares a una dada:

```
w2v_model.wv.most_similar('goal')
```

```
[('charlton', 0.9230924844741821),
 ('goalkeep', 0.9164312481880188),
 ('chanc', 0.9123638272285461),
 ('ronaldo', 0.9075625538825989),
 ('rooney', 0.9061418771743774),
 ('free', 0.8961011171340942),
 ('almunia', 0.8929105401039124),
 ('equalis', 0.8916003108024597),
 ('neil', 0.8858895897865295),
 ('cech', 0.8771913647651672)]
```

-Vector de una palabra:

```
w2v_model.wv['coach']
```

```
array([ 0.11675355, -0.130342  ,  0.4800894 ,  0.2641824 ,  0.3583388 ,
       -0.12193884, -0.46583775,  0.50676495, -0.1661445 ,  0.24802336,
       -0.18113795, -0.05517568, -0.26029655,  0.31659916, -0.0256466 ,
       -0.30430102, -0.37336907, -0.11424797,  0.19862047, -0.15398662,
        0.01242089, -0.16943517, -0.14069588, -0.04364046,  0.1379194 ,
       -0.03337369, -0.05080909,  0.10978305, -0.17698675, -0.04985935,
        0.10331126, -0.11608335, -0.31617883, -0.15831542, -0.05071599,
        0.1580947 , -0.04510977,  0.16157055,  0.1610449 , -0.25386176,
       -0.08722269, -0.07846735, -0.1743966 ,  0.2650097 ,  0.25265464,
       -0.11747578, -0.26479214, -0.09655338,  0.18463862,  0.3614522 ,
        0.13035935,  0.06031863, -0.16967298,  0.18282244,  0.16870807,
        0.02728668,  0.28039742, -0.29420322, -0.02653305, -0.05173177,
       -0.29022357,  0.45853758, -0.02665439,  0.23703094,  0.07138706,
        0.16010894,  0.02719226,  0.01089154, -0.04527301,  0.16560958,
       -0.01671212,  0.01715295, -0.00577979, -0.03251279, -0.04220404,
       -0.06574003, -0.08782996, -0.15415496, -0.2730347 ,  0.19933277,
        0.00967726,  0.04351076, -0.10994855,  0.17052756, -0.04720717,
       -0.14347805,  0.0266029 ,  0.2978253 , -0.03151633,  0.3004917 ,
       -0.02158401,  0.04960968, -0.05958434,  0.17024456,  0.02379286,
       -0.14726613,  0.02137849, -0.1365725 , -0.08144067,  0.17331323,
       -0.14704032,  0.44378203,  0.23839316, -0.50357395, -0.12911353,
       -0.20687872,  0.01741337,  0.02890475,  0.22517821, -0.11693526,
       -0.0848752 , -0.05489323,  0.30058476,  0.352287  ,  0.24023579,
       -0.10824311, -0.06983276,  0.08171532,  0.00356329,  0.33318213,
       -0.0535867 ,  0.09995777,  0.0749827 ,  0.22947511, -0.04271644,
        0.08900353,  0.07237633,  0.08166519,  0.14790882,  0.07315239,
        0.11123287, -0.1802867 , -0.03300031, -0.15382922,  0.11249683,
        0.24635984,  0.22510695, -0.11341313,  0.03483933, -0.20538607,
        0.01751183, -0.04693484, -0.14418186, -0.0758235 , -0.0183712 ,
       -0.0005652 , -0.16191164,  0.02167483, -0.21908036, -0.08203193,
       -0.0734763 , -0.1615121 , -0.12199818,  0.02662996, -0.07852374,
        0.33110878,  0.2864545 ,  0.20477277,  0.32385543,  0.02120322,
        0.04583699,  0.12863526, -0.37782267,  0.17909403, -0.12882097,
        0.02804962,  0.0198708 , -0.4923038 ,  0.09156771,  0.24275075,
       -0.2173734 ,  0.01550839, -0.30209014, -0.47190794,  0.16597474,
        0.346372  ,  0.22618409, -0.06193469,  0.03051975,  0.22121292,
       -0.06131676, -0.1625987 , -0.26288807,  0.27015433,  0.00213973,
       -0.05447162, -0.27553672,  0.2709396 ,  0.16917203,  0.13002142,
        0.12004394, -0.04904453, -0.06799144, -0.27378651,  0.06697614,
        0.07825035, -0.01712921,  0.20990832,  0.17733419,  0.17568038],
      dtype=float32)
```

-Palabra que menos relación tiene con el conjunto:

```
w2v_model.wv.doesnt_match(['arsenal', 'run', 'chelsea'])
```

```
'run'
```

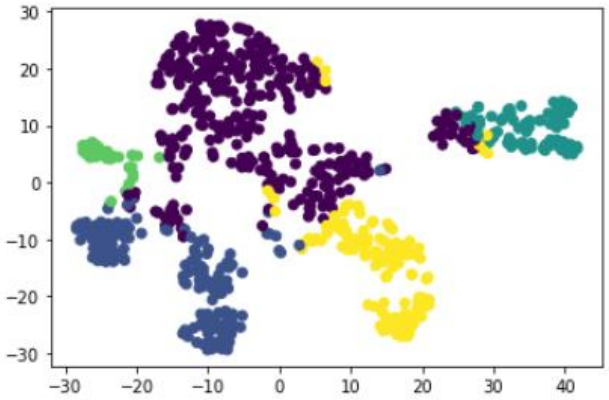
-Palabras pararecidas a otra:

```
w2v_model.wv.most_similar('arsenal')
```

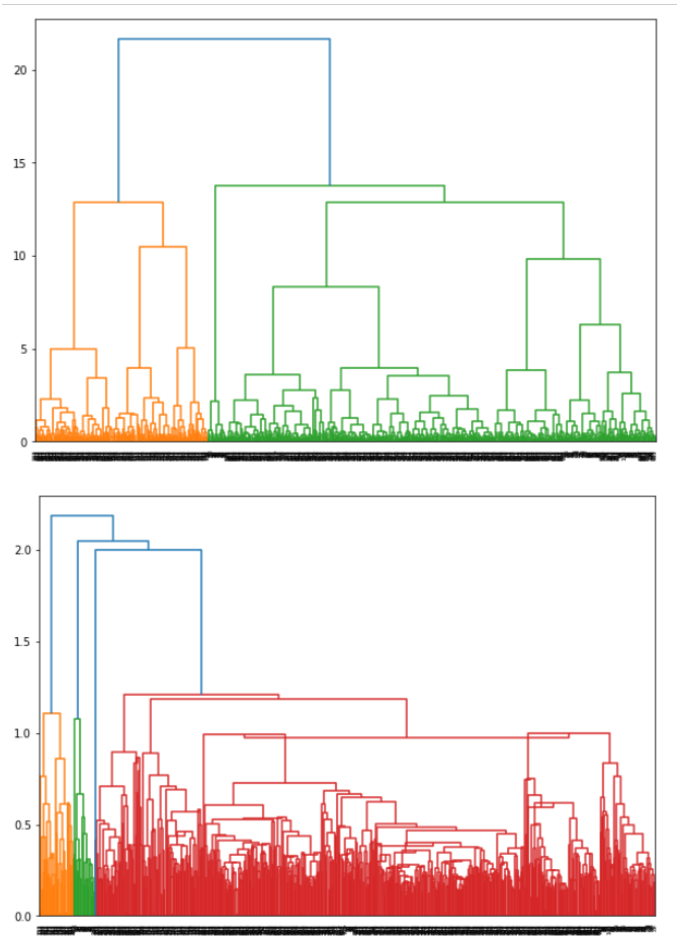
```
[('chelsea', 0.968766450881958),
 ('gunner', 0.9570778012275696),
 ('unit', 0.9549037218093872),
 ('striker', 0.954119086265564),
 ('portsmouth', 0.9515930414199829),
 ('everton', 0.9382190704345703),
 ('manchest', 0.9363642334938049),
 ('mourinho', 0.9293633699417114),
 ('boss', 0.9265363216400146),
 ('arsene', 0.9261350631713867)]
```

5.- Clustering: A la hora de aplicar clustering, hemos utilizado dos técnicas diferentes, la primera llamada KMeans y el algoritmo de clustering jerárquico aglomerativo.

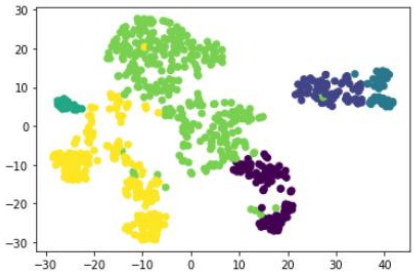
Para la técnica de KMeans, visualizamos los resultados utilizando la técnica de reducción de dimensionalidad t-SNE, la cual nos arroja el siguiente agrupamiento:



En el algoritmo de clustering jerárquico aglomerativo visualizamos los resultados utilizando un dendrograma, el cual puede variar dependiendo del método de enlace que usemos. Nosotros lo hemos probado con ‘ward’ y con ‘centroid’, dándonos los siguientes dendogramas respectivamente:



Luego cortamos el dendograma a un nivel específico y se asigna una etiqueda de cluster a cada documento. Luego visualizamos los resultados en un gráfico de dispersión:



V. Conclusiones

Como conclusión obtenemos ciertas ideas que han sido claves en el desarrollo del proyecto.

Procesamiento y tratamiento de los datos como parte fundamental del proyecto. La correcta tokenizacion, eliminación de stop words y stemming del proyecto es capaz de aligerar la carga computacional en gran medida, mejorando así la eficiencia del sistema. No hay comparación entre el procesamiento de los datos en crudo, que procesar el texto tras haberlo tratado de manera optima

Clustering como una óptima forma de clasificación de documentos. Los resultados obtenidos por los gráficos nos muestran que los grupos creados guardan relación con las etiquetas originales.

Referencias

[1] Página web del curso IA de Ingeniería del Software. https://www.cs.us.es/cursos/iais.

[2] Scikit-Lean Documentation https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

[3] Wikipedia.org

[4] Documentacion online sklearn https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html

[5] Chat GPT https://chat.openai.com/