Centro de Investigación en Matemáticas (CIMAT). Unidad Monterrey

Análisis de datos complejos: Tarea 3

Jorge Luis Ramos Zavaleta

15 de Octubre de 2018

1. Ejercicio

Este ejercicio es sobre clasificación y análisis de sentimientos. Considera los datos que se encuentran en spanish_reviews.zip, que corresponden a opiniones de usuarios en los siguientes productos: automóviles, hoteles, lavadoras, libros, teléfonos celulares, música, computadoras y peliculas. Para tu comodidad, he preparado dos conjuntos de datos: train (80 %) y test (20 %). Para cada uno de ellos hay dos categorias: yes y no, que indican las opiniones positivas y negativas, respectıvamente.

1. Implementa el clasificador ingénuo Bayesiano para las opiniones positivas y negativas. Usa los datos train y test para ajustar y verificar los resultados de tu clasificador, respectıvamente.

2. Ajusta clasificadores basados en SVM y otro de tu preferencia (CART, Boosting, NNet, etc...) usando la matriz de términos pesada con el criterio Term Frequency-Inverse Document Frequency (TF-IDF). Compáralo con el ingénuo Bayesiano.

3. Obten representaciones vectoriales de las palabras en los documentos de las opiniones usando word embeddings con word2vec. Verifica si existen patrones evidentes entre palabras que corresponden a adjetivos, sustantivos y/o diferentes entidades encontradas. Describe cómo podrıas usar esto para mejorar el clasificador ingénuo Bayesiano.

1.1. Solución

1. El clasificador ingénuo Bayesiano es considerado como un algoritmo de base para probar la eficiencia de otros algoritmos de clasificación ya que su desempeño tiende a mostrar un buen resultado aun cuando se trata de un algoritmo bastante simple.

   En nuestro caso usando las matrices de frecuencias de los datos de entrenamiento obtuvimos lo siguiente

```
table(clas.orig.train,clas.est.train)  #Tabla de confusion de la
                                       # prediccion del entrenamiento
               clas.est.train
clas.orig.train   N   P
              N  95  65
              P   7 153
```

   Con un nivel de precisión de 77.5 % para dichos datos, observando que su principal fallo se da al clasificar documentos que en realidad estan catalogados como negativos pero son catalogados como positivos. Para el caso de los datos de prueba se obtuvo lo siguiente

```
              clas.est.test
clas.orig.test  N  P
             N 21 19
             P 11 29
```

   Con un nivel de precisión de 62.5 %, en este caso podemos ver que aunque clasifica mal los documentos que son negativos como positivos como en los datos de entrenamiento, el caso analógo tambien tiene un gran numero de casos.

2. Para contrastar este resultado se considero usar la matriz de términos pesada usando el criterio Term Frequency-Inverse Document Frecuency (TF-IDF) usando otros 3 clasificadores.

   Para realizar las pruebas se etiquetaron los datos como -1 para negativo y 1 para positivo. La primera prueba se hizo utilizando SVM con un kernel de tipo sigmoid con lo que se obtuvo una clasificación buena como lo muestra la siguiente tabla de confusión

```
          pred
         -1  1
Orig  -1 35  5
       1  11 29
```

Con un nivel de precisión de 80 %. Ahora usamos un arbol de clasificación con profundidad máxima de 30, usando 10 dobleces para la validación cruzada y con parámetro de complejidad igual 0.05, con esta configuración se obtuvo la siguiente tabla de confusión para los datos de prueba

```
table(etiquetas[1:80],pred2)
    pred2
     -1  1
  -1 34  6
  1  21 19
```

Con un nivel de precisión reportado de 66.25 %, y en este caso y en el de SVM se puede ver el efecto contrario de clasificar mas documentos catalogados como positivos como negativos al contrario del método de Bayes.

Por último aplicamos una red neuronal de una capa oculta de tamaõ uno, con lo que se obtuvo la siguiente tabla de confusión

```
table(etiquetas[1:80],pred3)
    pred3
     -1  1
  -1 21 19
  1   6 34
```

con un nivel de precisión de 68.75 %, y catalogando más documentos negativos como positivos como fue el caso de Naive Bayes.

En los tres casos se reporta un nivel de precisión mas alto que con el método de Bayes como era de esperarse, considerando que Bayes es nuestro algoritmo base.

3. Como parte del proceso se establecio una representación de las palabras usando el algoritmo **Word2Vec** con el fin de entrever posibles relaciones basadas en la semátinca de los documentos.

   En las figuras 1.1 a 1.4 se encuentran algunas relaciones que pueden considerarse especiales dada la semántica de los documentos. En la figura 1.1 se tiene una representación en las primeras dos componentes de las palabras mas cercanas a las palabras: *bien, buena, bueno, confianza*. En este grafico se pueden apreciar palabras cercanas como necesario, viable y costo, que parecen tener bastante relevancia con respecto a las palabras que elegimos y que pueden considerar palabras asociadas a un sentimiento positivo.

Después se hizo la busqueda de relaciones con respecto algunos de los articulos sobre los que se hacen las reseñas, en el caso de pelicula en la figura 1.2 se pueden apreciar palabras como cinta, industria, comedia y actriz. Para el caso de la palabra computadora en 1.3 aparecen palabras cercanas y relevantes como hp, acer, y memoria RAM.

En la figura 1.4 para la palabra lavadora se observan palabras como 7 kg, carga, ropa, detecta, seca, que ciertamente pueden ser consideradas como palabras con una semántica cercana a la palabra lavadora. Por último en la figura 1.5 se observan las palabras cercanas a la palabra coche, y es relevante encontrar la palabra seguridad como una característica importante para un coche, tambien se encuentran las palabras fiat, concesionario, km, vehiculo y viajar que ciertamente tienen una relación directa con la palabra coche.

Dado que usando el método de inmersión **Word2Vec** se pudieron encontrar relaciones significativas entre las palabras, uno tiene la necesidad de usar estas nuevas características para mejorar la precisión de un modelo. En el caso de Bayes una posibilidad es usar algun método de cluster sobre las palabras vectorizadas estableciendo relaciones y usando cada cluster como entrada para el método de Bayes, esto nos permitiría hacer uso de la semántica de las palabras, esto siempre que se use un método de clustering basado en distancias que en este caso sería usar una métrica de tipo correlación.

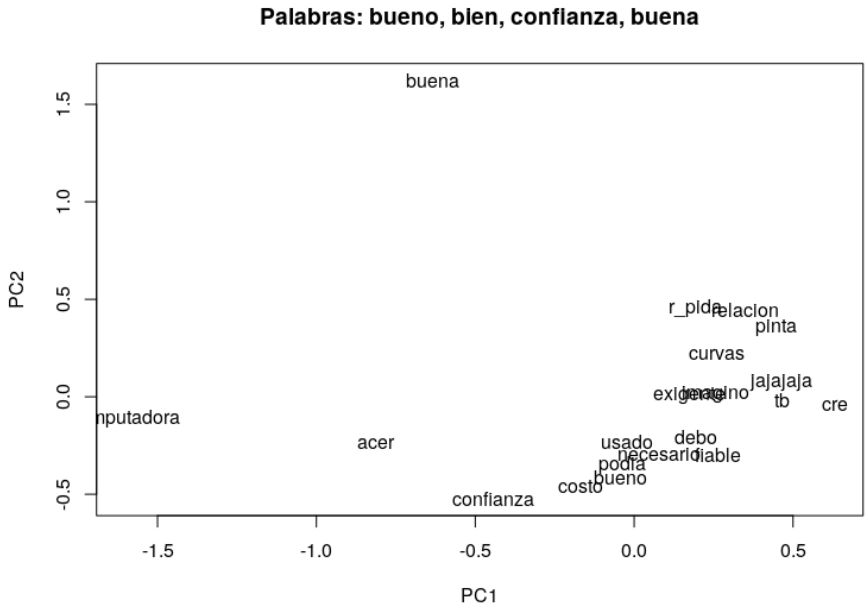

Figura 1.1: Representación en las primeras 2 componentes principales de las palabras cercanas a bueno, bien, confianza y buena

2. Ejercicio

Este ejercicio es sobre semántica vectorial y word embeddings. En el archivo *spanish_billion_words.zip* se encuentra un Corpus en español que reúne corpus de diferentes fuentes.

1. Utiliza word2vec para obtener representaciones semánticas vectoriales de las palabras del corpus. Verifica cualitativamente su desempeño escogiendo algunas palabras clave. Estas pueden ser arbitrarias o corresponder a ciertas etiquetas gramaticales. Indica el criterio que usaste para seleccionarlas. Usa gráficos informativos para ilustrar tus respuestas.

2. Realiza clustering mediante Kmeans. Elige alguno(s) valor(es) K, Ilustra y comenta tus hallagos.

2.1. Solución

1. El dataset *spanish_billion_words.zip* contiene 100 documentos con diversas palabras en español. Sin embargo, debido a cuestiones de tiempo de cómputo solo se consideraron 3 documentos entrenados usando bigramas para encontrar posibles entidades o

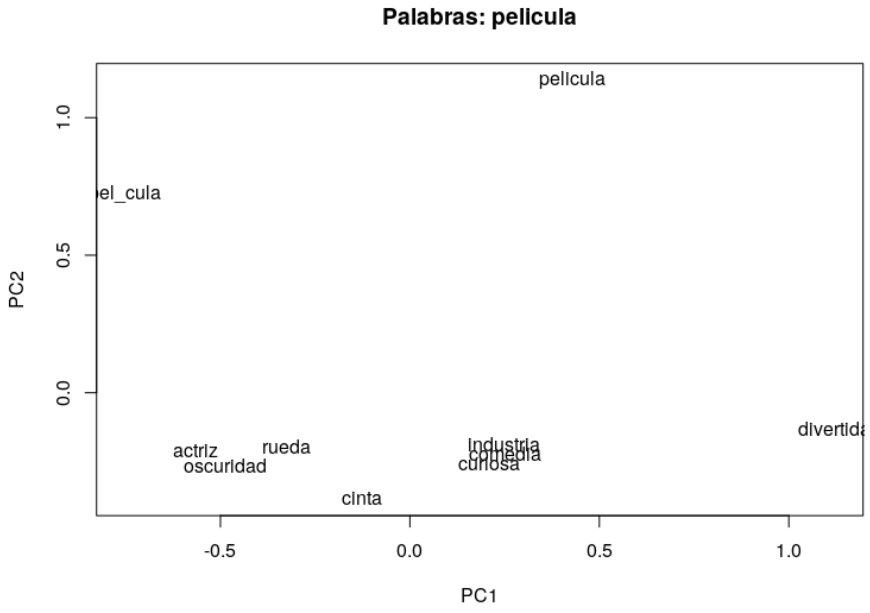

Figura 1.2: Representación en las primeras 2 componentes principales de las palabras cercanas a la palabra pelicula

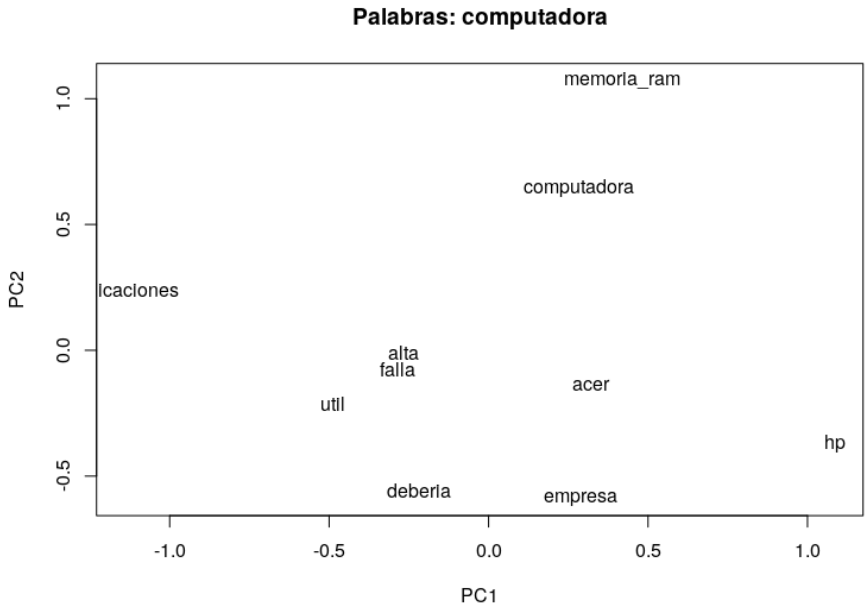

Figura 1.3: Representación en las primeras 2 componentes principales de las palabras cercanas a la palabra computadora

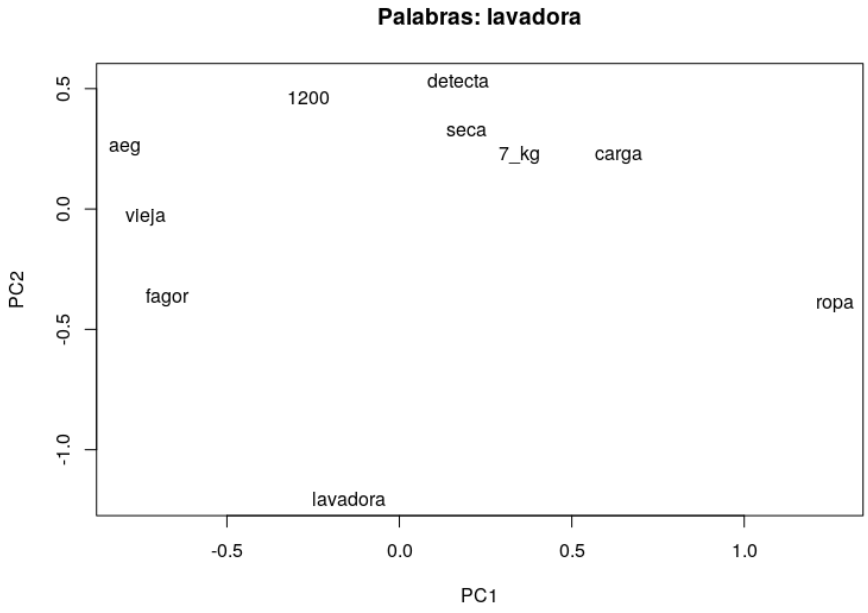

Figura 1.4: Representación en las primeras 2 componentes principales de las palabras cercanas a la palabra lavadora

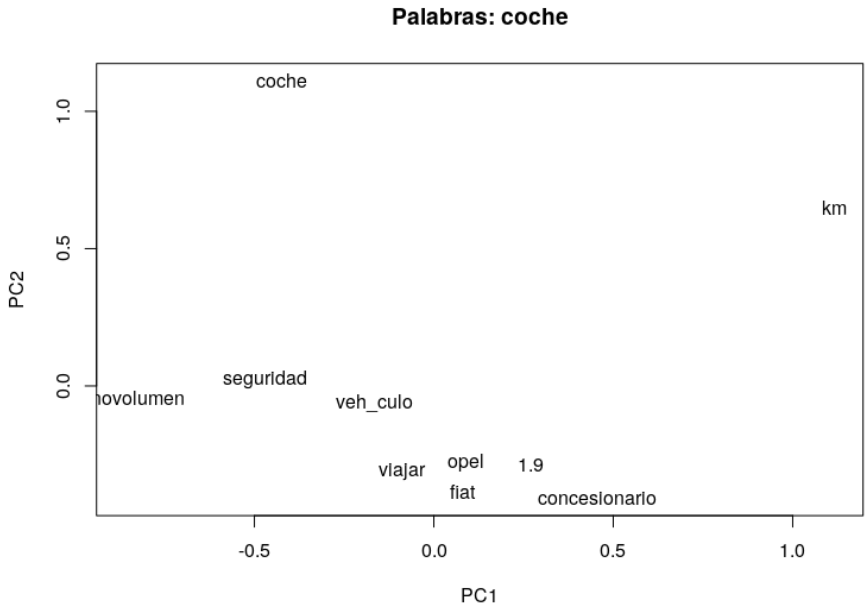

Figura 1.5: Representación en las primeras 2 componentes principales de las palabras cercanas a la palabra coche

palabras que regularmente van juntas como pueden ser desarrollo económico o política comercial. Los documentos usados fueron los etiquetados como 17,18 y 19, y se generaron 200 vectores.

Debido a algunos problemas con la libreria no se generaron graficos usando PCA como en el ejercicio anterior, sino que se consideraron las palabras mas cercanas haciendo uso de la distancia coseno. Para la verificación cualitativa del desempeño del modelo **Word2Vec** se hizo un grafico con todas las palabras que se muestra en la figura 2.1 y a partir de ahí se eligieron algunas palabras que parecian interesantes con respecto al contexto de los documentos.

Primero se eligio un conjunto de palabras laboral, trabajo, empresa y revisamos la cercania con las 7 palabras mas cercanas a ellas. En la figura 2.2 se puede observar su representación con respecto a la distancia coseno. Aqui el bigrama mas cercano es rubén urbano que no tiene ninguna relevancia directa con las palabras, todas las palabras cercanas se presentan a continuación

`trabajo, laboral,empresa, rubén_urbano, mujeres_solteras, costigliolo, trabajadora, además, temporarios, gabriel_morcelli`

viendo que si hay algunas palabras con semántica significativa como mujeres solteras, temporarios y trabajadora. Después se ubicaron las 8 palabras mas cercanas a los bigramas desarrollo y desarrollo_económico, lo cuales se pueden ver en la figura 2.4, aqui puede observarse mucho mas concordancia con las palabras, las palabras encontradas son

`desarrollo_económico, desarrollo, fomento, impulsar, progreso_económico, fomentar, fortalecimiento_institucional, bndes, crecimiento_económico, integración_regional`

Aqui se pueden observar bigramas muy relevantes como progreso económico, fortalecimiento institucional, crecimiento econmíco e integración regional, en general todas estas palabras tienen una relación estrecha en algún contexto de discurso político-económico. Por último se eligieron las palabras política y confrontación, y se buscaron las 8 palabras mas cercanas a ellas. En la figura 2.1 se muestra graficamente la relación de dichas palabras. Las palabras encontradas son

`confrontación, política, ideológica, varias_lecturas, estorba, crítica, actitud, españa_plural, bajar_tensiones, convulsionada`

En este sentido los bigramas crítica, actitud, españa cultura, bajar tensiones, pueden ser bastante significativos con respecto a las palabras usadas, aunque otros bigramas como varias lecturas y convulsionada, pueden referirse a un ambiente acádemico en donde se esten tratando estos temas, por lo que no debe descartarse su significancia.

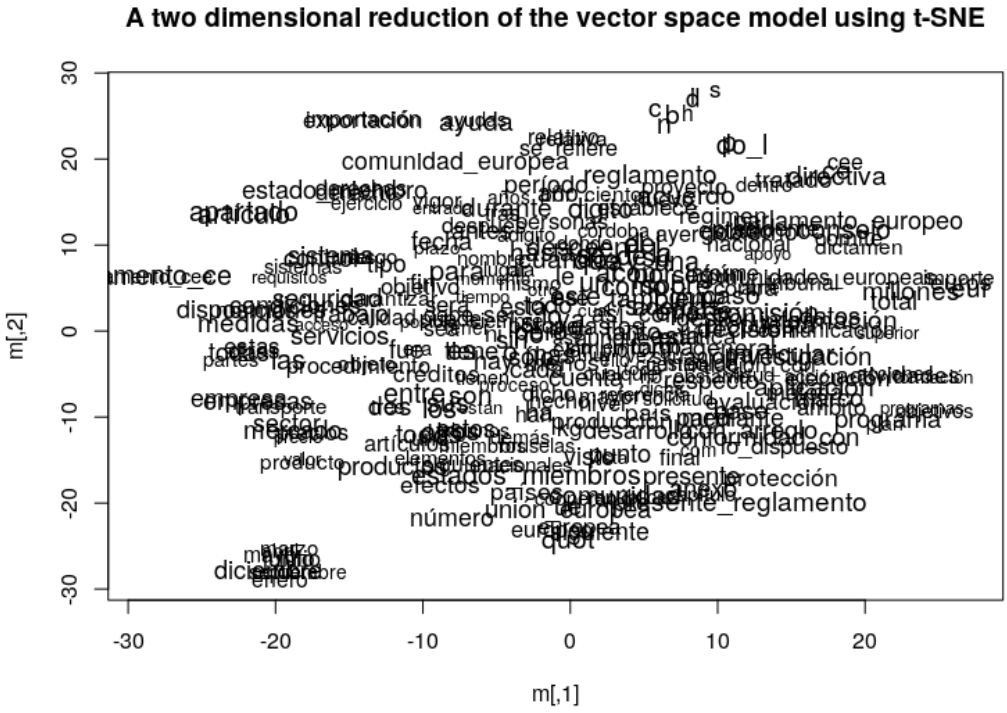

Figura 2.1: Representación en las primeras 2 componentes principales de las palabras vectorizadas en spanish billion words

2. Para la siguiente parte se realizo kmeans considerando 10 y 5 clusters para intentar encontrar algunas relaciones dentro de dichos clusters. Debido al gran numero de palabras solo se muestran las primeras 20 en algunos clusters donde se considero que existia relevancia. Primero aplicando con 10 clusters se encontraron los siguientes 3 clusters relevantes

```
> names(clustering1$cluster[clustering1$cluster==3][1:20])
"artículo", "reglamento_ce",  "apartado", "p", "do_l", "ce",
"directiva", "presente_reglamento", "reglamento", "eur",
"conformidad_con", "con_arreglo", "anexo", "estado_miembro",
"parlamento_europeo", "período", "disposiciones", "gastos"
"comunidad_europea", "siguiente"
```

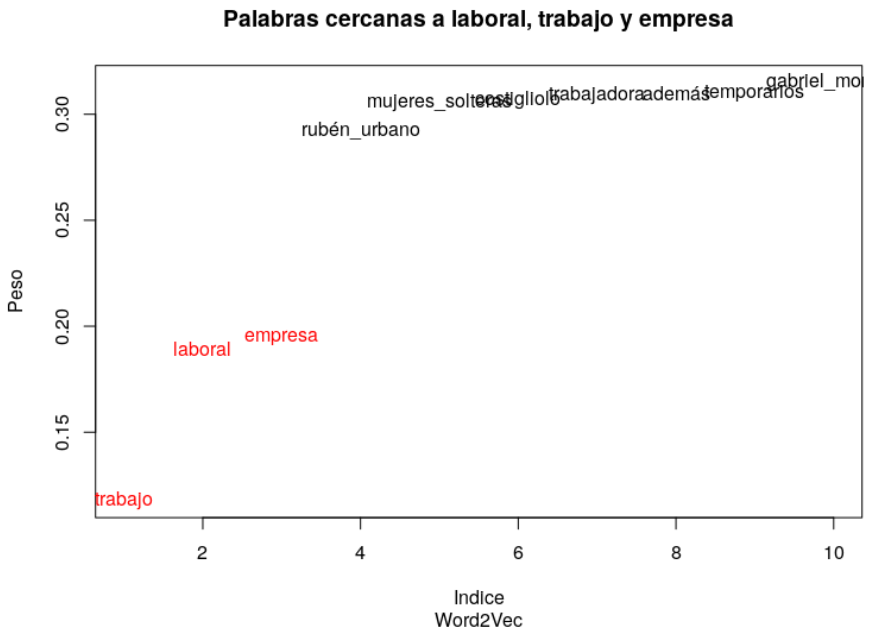

Figura 2.2: 7 Palabras mas cercanas a laboral, trabajo, empresa con respecto a la distancia coseno

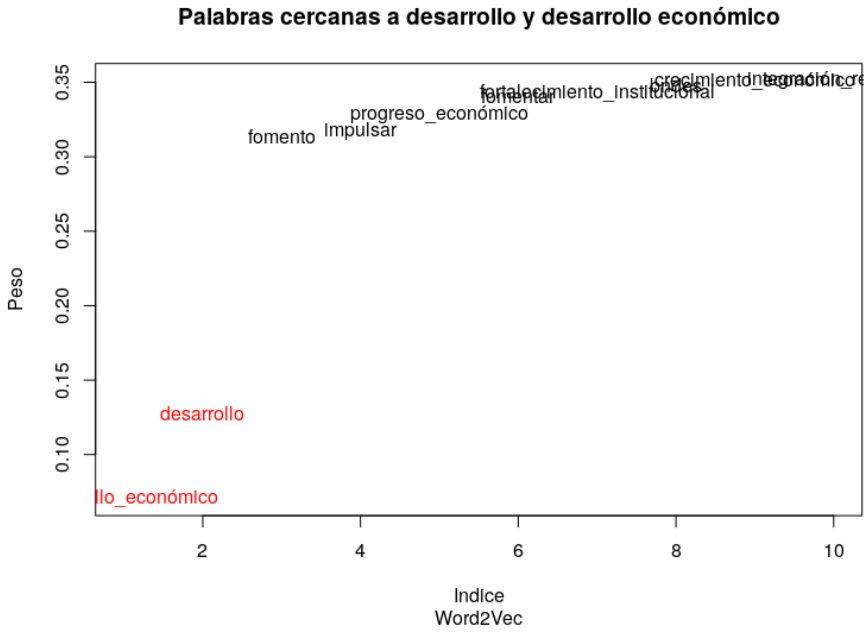

Figura 2.3: 8 Palabras mas cercanas a desarrollo y desarrollo_económico con respecto a la distancia coseno

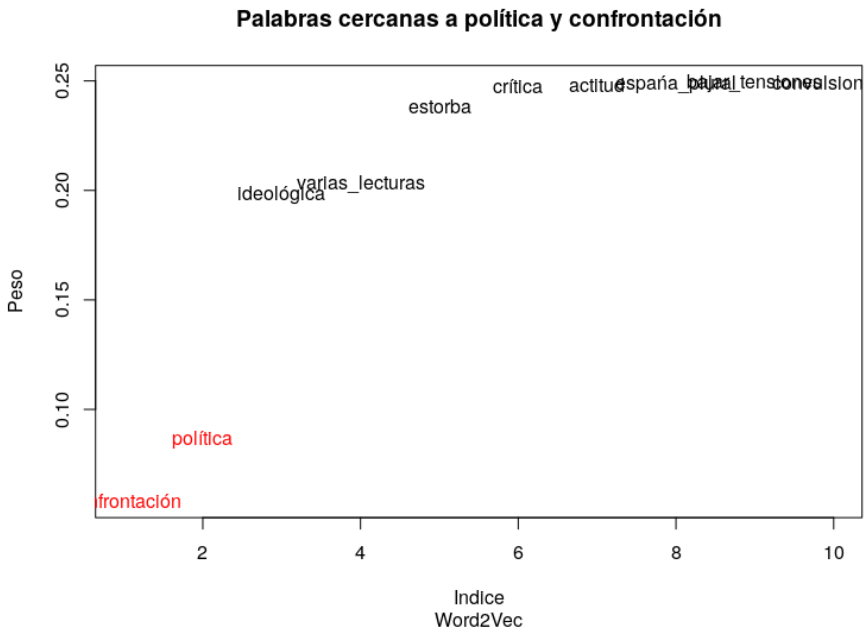

Figura 2.4: 8 Palabras mas cercanas a política y confrontación con respecto a la distancia coseno

```
> names(clustering1$cluster[clustering1$cluster==4][1:20])
"adigito", "fabricación", "cdigito", "cereales", "cen", "exdigito", "mm",
"ensayo", "iec", "leche", "materias", "aparatos", "máquinas", "hortalizas",
"iso", "aceite", "mantequilla", "fábrica", "sdigito", "acero"

> names(clustering1$cluster[clustering1$cluster==8][1:20])
"digito", "de", "la", "el", "en", "y", "a", "que", "los", "del", "las",
"se", "por", "para", "un", "no", "una", "con", "al", "su"
```

Cabe recordar que solo son las primeras 20 palabras las que se consideraron por lo que puede existir algun otro cluster relevante pero que la relevancia puede iniciarse después de la palabra 20. El primer cluster puede hacer referencia a cuestiones tratados o leyes, el segundo a cuestiones de manufactura de materias primas, y el tercero a palabras que pudimos eliminar para mejorar nuestro preprocesamiento como conjunciones.

Ahora se realizo kmeans para ubicar 5 cluster y de igual manera solo se consideraron las primeras 20 palabras encontradas.

```
names(clustering2$cluster[clustering2$cluster==1][1:20])
"digito", "de", "la", "el", "en", "y", "a", "que", "los", "del", "las",
"se", "por", "para", "un", "no", "una", "con", "al", "o"

> names(clustering2$cluster[clustering2$cluster==3][1:20])
"and", "the", "in", "of", "et", "of_the", "to", "in_the", "à", "as",
"eu", "for", "on", "is", "du", "commission", "or", "by", "and_the",
"to_the"

> names(clustering2$cluster[clustering2$cluster==4][1:20])
"</s>", "artículo", "reglamento_ce", "apartado", "p", "estados_miembros",
"do_l", "ce", "aplicación", "quot", "directiva", "presente_reglamento",
"reglamento", "medidas", "eur", "conformidad_con", "productos",
"con_arreglo", "anexo", "unión_europea"

> names(clustering2$cluster[clustering2$cluster==5][1:20])
"kg", "adigito", "fabricación", "x_x", "cdigito", "cereales", "excepto",
"cen", "exdigito", "eur_t", "mm", "ensayo", "kg_ldigito", "iec", "leche",
"materias", "aparatos", "máquinas", "modificada", "código_nc"
```

El primer, tercero y cuarto clusters se parecen mucho a los interesantes que se encontraron con 10 clusters, aunque algo interesante es que el segundo cluster encontró

palabras en inglés dentro del documento y las agrupo todas.

3. Ejercicio

Este ejercicio es sobre Machine Translation (MT).

El objetivo es implementar en forma simplificada, el paper de Mikolov et al., el cual, a grandes rasgos, consiste en extender de forma automática diccionarios que puedan traducir palabras y frases de diferentes lenguajes usando representaciones vectoriales de palabras dentro de dichos lenguajes.

El método es sencillo y sus pasos se describen a continuación.
Considera un problema de traducción del un lenguaje fuente a otro lenguaje objetivo.
En este ejemplo será del español al inglés.

- Obtén embeddings monolinguales usando word2vec sobre un corpus adecuado para cada idioma, obteniendo ası espacios vectoriales para el lenguaje fuente A y el objetivo B.
- Usa un diccionario bilingüe reducido para obtener un mapeo lineal entre los es- pacios vectoriales de ambos idiomas a través de una Matriz de Traducción W, que obtienes resolviendo

$$\min_{\mathbf{W}} \sum_{i=1}^{n} ||\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{z}_i||^2,$$

  donde $\{\boldsymbol{x}_i, \boldsymbol{z}_i\}_{i=1}^{n}$ son los embeddings de los pares de palabras del diccionario que traducen $\boldsymbol{x}_i \rightarrow \boldsymbol{z}_i$. Estos los seleccionas de los embeddings que corresponden a los corpus monolinguales.
- Para una nueva palabra $x \in \mathbb{A}$, obten su mapeo mediante $\boldsymbol{z} \approx \boldsymbol{W}\boldsymbol{x}$, y obtén su traducción correspondiente buscando su embedding más cercano (distancia coseno) en el espacio del lenguaje objetivo $\mathbb{B}$.

Implementa el procedimiento anterior usando los corpus que se encuentran en el archivo europarl_es-en.zip, que contiene corpus paralelos (español-inglés) extraídos de reuniones y debates del parlamento europeo. Los corpus paralelos son versiones alineadas en dos idiomas de un mismo documento. Verifica el desempeño del método con algunos ejemplos ilustrativos de palabras como en el paper de Mikolov. ¿Son equivalentes los resultados? ¿Qué sugieres para mejorarlos?

3.1. Solución

Se realizo la implementación del algoritmo del paper de Mikolov usando un diccionario paralelo de traducción español-inglés correspondientes a reuniones y debates del parlamento europeo. Para ello se uso api de Google Traslate para generar una traducción directa de las 5,000 palabras mas frecuentes en el idioma español. Se consideraron 150 vectores en el embedding.

Para probar el algoritmo se muestran algunos ejemplos, el primero de ellos es para la palabra derecho que nos muestra los siguientes resultados

1	right	0.7228216
2	rights	0.6808323
3	inalienable_rights	0.6300018
4	every_person	0.6112689
5	freedom	0.6088840
6	liberties	0.6017805
7	express_one's	0.6003054
8	inalienable_right	0.5955537
9	an_inalienable	0.5923615
10	each_person	0.5919239

donde se muestra que en general es un buena traducción. Para los siguientes ejemplos también se logro una buena traducción los bigramas usados son derechos humanos y medio ambiente respectivamente

1	human_rights	0.8821517
2	democratic_freedoms	0.7848910
3	defenders	0.7783254
4	fundamental_freedoms	0.7619571
5	religious_freedom	0.7559659
6	violations	0.7487011
7	civic_freedoms	0.7166575
8	democratic_principles	0.7126398
9	liberties	0.7031403
10	minority_rights	0.6904916

1	environment	0.8632792
2	environmental_protection	0.7809087
3	environmental	0.6754283
4	health	0.6583009
5	biodiversity	0.6573010
6	people's_health	0.6446129

```
7       natural_environment                  0.6139349
8             public_health                  0.6090719
9             bio_diversity                  0.6023666
10                  ecology                  0.5992418
```

Un extraño caso de traducción sucedio al introducir 2 palabras que aun cuando estaban en el corpus en el español la traducción esta completamente errónea aun cuando su distancia coseno es relativamente alta. Se eligieron las palabras borracho y compita y se obtuvieron los siguientes resultados respectivamente

```
1          cellar                  0.7365536
2           beard                  0.7277446
3        chatting                  0.7244864
4       briefcase                  0.7194503
5         virgins                  0.7181672
6       pestering                  0.7146906
7  walking_around                  0.7146223
8             het                  0.7112671
9             mum                  0.7086339
10        jackets                  0.7028325
```

```
1          compete_with                  0.6685056
2               compete                  0.6361383
3    competitive_battle                  0.6006589
4              competes                  0.5918911
5             competing                  0.5698778
6              tap_into                  0.5665783
7  fiercely_competitive                  0.5635461
8        compete_fairly                  0.5635121
9     truly_competitive                  0.5503059
10          competitive                  0.5384211
```

en ambos casos se puede observar que ninguna de las posibles traducciones tiene nada que ver con las palabras elegidas.

Una de las cosas que puede hacer para mejorar la traducción es hacer uso de un diccionario de palabras más grande que permita establecer un espacio mas amplio de búsqueda. Otra posibilidad es generar varios diccionarios para establecer diccionarios especializados donde se puedan encontrar palabras en su contexto específico y generar una mejorar traducción, esto implica tener varias matrices de traducción y se debe establecer una forma catalogar el documento de manera inicial.