Detección automática de tópicos en el Diario de los Debates

Christian Cuéllar Pérez Rea
Stephane Keil Ríos

Minería de Texto
Instituto Tecnológico Autónomo de México

1 Introducción

La generación de grandes volúmenes de texto hace prioritario el desarrollo de herramientas que permitan recuperar de manera automatizada la información más relevante con la finalidad de facilitar el análisis de contenido. El campo de la minería de texto se basa en la idea de extraer información de alta calidad derivada del análisis de tendencias y patrones encontrados en formatos de texto no estructurados.

Sin embargo, este tipo de información por lo general viene en formatos no estructurados que hacen que la aplicación de algoritmos de aprendizaje de máquina difícil. En efecto muchos de estos algoritmos requieren que la información de entrada sea numérica, es decir que los textos no estructurados sean transformados a información estructurada. En este sentido uno de los grandes retos es el análisis automatizado de la semántica, es decir el significado, de los textos que se estudian. La representación matemática de esta información no estructurada permite realizar ciertos tipos de análisis muy poderosos, como por ejemplo, el agrupamiento de documentos que contienen información que tratan de temas semejantes o la identificación de los términos multipalabra que caracterizan cierto tipo de documentos. Las técnicas modernas de minería de texto han tenido avances importantes en permitir a la computadora encontrar el significado de palabras y documentos sin la intervención del ser humano. Un caso muy sonado es el uso de redes neuronales profundas (word2vec) que encontraron que la relación de hombre a mujer es la misma que la de rey a reina.

En las siguientes líneas se propone una metodología que incorpora varios algoritmos de minería de texto y de aprendizaje de máquina con el objetivo de obtener los tópicos latentes en los documentos históricos del Diario de los Debates de la Cámara de Diputados en México. Estos diarios son publicaciones que reproducen el contenido de los debates públicos e incorporan los oficios, peticiones, iniciativas, votaciones y otros asuntos que fueron tratados en las sesiones de la Cámara baja. Es importante mencionar que los documentos

analizados contienen más de doce mil páginas de texto no estructurado, por lo que resulta evidente que este volumen de información requeriría de un inmenso esfuerzo para ser analizada por personas y, por lo tanto, se justifica el intentar aplicar técnicas modernas de minería de texto. La pregunta que guía nuestra investigación es básicamente: ¿De qué se habló en las sesiones de las Cámara baja en los periodos posteriores a las elecciones presidenciales de 1946 a 1976?

2 Objetivos

El objetivo del proyecto es el de analizar los tópicos que se discutieron en el Diario de los Debates de la Cámara de Diputados en periodos posteriores a las elecciones federales. Debido a que el texto corresponde a la transcripción textual de lo ocurrido en la Cámara de Diputados el corpus es muy variado: se pueden observar discursos en primera persona, transcripción de leyes que se discutieron, recuentos de hechos de manera narrativa, inclusive transcripción de temas de la proxemia en la Cámara, como lo es el caso de los Aplausos. El tratar con información tan diversa y con alto volumen complica en extremo el análisis de patrones en la información. Debido a la diversidad y volumen de esta información es necesario desarrollar una metodología que combine varios algoritmos de minería de texto para dividirlo en subdocumentos que traten de seguir los cambios de temática tratados en la Cámara, identificar cuáles son las temáticas recurrentes e identificarlas por sus términos clave para poder caracterizarlas. Una de las posibilidades es la de extraer los metadatos de las temáticas de mayor interés, como por ejemplo, temas de presupuesto, temas electorales, temas legislativos, entre otros.

México es un país con una difícil historia democrática, los procesos electorales se han visto manchados por ocurrencias de fraude electoral. Sin embargo, no existe ninguna base de datos estructurada que indique las ocurrencias de estos fraudes, fechas, ubicaciones geográficas y el tipo de irregularidad que se cometió. No obstante, esta información se puede encontrar potencialmente al revisar las transcripciones de lo ocurrido en el Diario de los Debates. Construir una base de datos de esta índole permitiría realizar un análisis profundo de las tendencias sobre ocurrencias de fraude electoral a través de la historia de nuestro país. Al profundizar en las ubicaciones geográficas donde ocurrieron fraudes, los partidos políticos y los candidatos involucrados y los tipos de fraude detectados se podría cruzar con los resultados de estas elecciones para encontrar patrones recurrentes de este tipo de incidencias.

El problema que buscamos resolver es desarrollar una metodología sencilla y reproducible para el análisis de grandes volúmenes de texto que abordan de manera repetitiva pero no ordenada las mismas temáticas. Ser capaces de extraer una serie de términos clave de cada una de estas temáticas para poder detectar de manera automatizada cuando el texto habla sobre alguno de estos tópicos y, de ser el caso, poder extraer información específica de alguna temática particular, como lo es el caso de las ocurrencias de fraude electoral.

3 Metodología

La metodología que proponemos se divide en cuatro grandes pasos que se pueden resumir de la siguiente manera:

1. Separar los documentos en subdocumentos que separen las temáticas que se debaten en la Cámara baja.

   Cada uno de los Diarios de los Debates corresponde a las transcripciones de múltiples días de sesión, en donde se abordaron una gran cantidad de temas, por lo que no existe un separador predefinido entre estos. La primera tarea fue encontrar un término de corte adecuado que permitiera separar estas temáticas. Aunque se consideró el uso de técnicas automatizadas para la detección de estas palabras de corte, debido al tiempo y al hecho de que uno de los miembros del equipo tenía experiencia previa en el análisis de los documentos, se tomaron términos conocidos para la separación de los archivos.

2. Representar los subdocumentos por sus características semánticas, terminología utilizada, coocurrencia de términos y, en general, elementos que permitan distinguir los temas que se discuten.

   Existen diferentes técnicas para la representación vectorial (numérica) de documentos. Por un lado, las técnicas de bolsa de palabras se basan en la frecuencia de aparición de ciertos términos y de su aparición mutua, sin embargo, tienen varias deficiencias ya que no consideran el orden de las palabras, ignoran la semántica y requieren que los documentos tengan aproximadamente la misma longitud. Recientemente han tomado fuerza otro tipo de algoritmos no-supervisados para la representación vectorial de palabras y documentos que superan las limitaciones mencionadas. Estos modelos de lenguaje utilizan redes neuronales y técnicas de reducción de la dimensionalidad para generar representaciones probabilísticas de las palabras en términos de los contextos en los que estas aparecen. Estos modelos han logrado que las representaciones vectoriales de palabras con contenido semántico similar sean, a su vez, similares. Un documento es al final una composición de palabras, por lo que si se combinan adecuadamente las representaciones vectoriales de las palabras que componen un documento, sería posible tener una representación vectorial de los documentos en donde aquellos documentos similares semánticamente tengan representaciones vectoriales parecidas. La técnica de paragraph vector y su implementación en doc2vec permite capturar la semántica general de documentos y, por lo tanto, estudiar que tanto se parece un documento a otro. En esta fase se obtiene una matriz numérica donde cada renglón representa un documento y las columnas son la representación vectorial de cada documento obtenida a partir de la composición de las palabras, su orden y su co-ocurrencia dentro de cada documento.

3. Agrupar todos los documentos que por sus características morfológicas y semánticas se pueden considerar similares.

Una vez que se tiene una representación numérica de los documentos es posible agruparlos entre sí de tal suerte que los grupos encontrados minimicen la varianza dentro de cada grupo y maximicen la varianza entre los grupos. Para esto es necesario el uso de algoritmos no supervisados o de agrupamiento. Estos algoritmos son capaces de agrupar los elementos de una colección en grupos diferentes entre sí. Decidimos para este ejercicio utilizar un algoritmo muy conocido llamado K-medias que de manera iterativa va encontrando los grupos al asignar cada elemento al centroide más cercano. Los centroides se recalculan en cada iteración tomando el centro de la hiperesfera de todos los elementos que lo componen. Este método tiene grandes deficiencias, una de ellas es la necesidad de determinar de manera heurística el número de grupos que se buscan encontrar. Sin embargo, es un método muy sencillo de implementar y sus resultados son fácilmente interpretables.

Otro método de agrupamiento que no requiere que el investigador determine de manera anticipada el número de grupos que espera encontrar es el algoritmo de MajorClust, el cual está enfocado a la categorización de documentos basado en técnicas de grafos. Este algoritmo de densidad representa a los documentos como nodos de un grafo y sus aristas son pesadas de acuerdo a una medida de distancia entre estos nodos; usualmente se utiliza la similitud por coseno en el caso de análisis de texto. MajorClust busca maximizar la conectividad total del grafo. Cada nodo inicialmente es un grupo separado y en cada iteración se le asigna al grupo hacia el cual tiene la menor distancia. Una de las grandes ventajas de este método es que no se debe dar a priori el número de grupos que se buscan encontrar en la colección, sino que el mismo algoritmo lo determina como parte de su proceso.

Aunque los dos algoritmos de agrupamiento son diferentes entre sí, ambos devolverán una cierta cantidad de grupos compuestos por documentos que deberían compartir características importantes entre sí. Normalmente se utilizan medidas estadísticas para caracterizar los grupos. Sin embargo, la representación vectorial por doc2vec hace la interpretación de los valores numéricos imposibles y es necesario encontrar otros mecanismos de caracterización.

4. Caracterizar los grupos de temáticas por términos multipalabra clave que permitan identificar las grandes temáticas que se discuten en la Cámara baja.

   Para caracterizar los grupos encontrados es posible realizar una exploración automatizada de términos que aparecen con regularidad en el texto. Los métodos de extracción de palabras clave buscan encontrar cuáles son las palabras o términos que resumen mejor un documento. Existen diversos algoritmos de extracción de palabras clave. Debido al volumen y diversidad de los documentos, nos enfocamos únicamente a métodos automáticos de detección. Los métodos tradicionales se basan en la ex-

tracción de palabras en función a su frecuencia y regularmente relativo a la frecuencia de aparición en un corpus dado. Estos enfoques tienen ciertas limitaciones, por ejemplo, sólo detectan palabras que ocurren más frecuentemente en un documento respecto al resto. En nuestro caso, dado que nuestro corpus está compuesto únicamente por el diario de los debates, decidimos utilizar métodos de extracción basados en documento (y no en corpus) que además permitan la extracción de términos multipalabra. El método RAKE (Rapid automatic keyword detection) nos brinda una manera sencilla y rápida de cumplir con nuestro cometido. Este método requiere del uso de palabras de paro, que no aparecen en lo que consideramos términos clave de un documento. Se generan secuencias de palabras entre estas palabras de paro y son candidatas aquellas secuencias cuyas palabras regularmente aparecen en conjunto en el documento. Las palabras y términos candidatos se califican en función a esta idea y se reportan al usuario.

5. Determinar cuáles de los términos multipalabra clave son únicos y permiten distinguir una temática de las demás.

   Durante las pruebas realizadas inicialmente, descubrimos que debido a la naturaleza de los documentos, las palabras clave extraídas podían repetirse a lo largo de varios grupos, además de que ciertos términos que nos parecen de particular interés no siempre eran clasificados entre los más altos. Decidimos entonces realizar la extracción de palabras clave sobre estas palabras clave ya extraídas utilizando un método de extracción basado esta vez en corpus. Cada grupo es entonces un documento y se evalúan las palabras clave extraídas del algoritmo RAKE usando la técnica de tf-idf donde se reclasifican los términos clave comparados con su aparición en los demás grupos. De esta manera se obtuvieron los términos clave que identifican de manera única a cada uno de los grupos, permitiendo así una clasificación más sencilla.

4 Resultados

1. Separar los documentos en subdocumentos que separen las temáticas que se debaten en la Cámara baja.

   El primer reto de la presente investigación fue el tratamiento de la documentación fuente. Cada uno de los documentos de los seis años para los que se tenían los diarios (1946, 52, 58, 64, 70 y 76) fue transformado a un archivo de texto. Después de realizar un análisis exploratorio de su contenido, se identificó que las palabras “Honorable Asamblea:” y “Honorable asamblea:” marcaban una separación clara entre las temáticas de los párrafos previos y los siguientes, por lo que se decidió utilizar como delimitador para generar los documentos que asumimos tratan de temas diferentes. Esto es lógico ya que cada vez que un diputado se dirige a la Cámara generalmente inicia con estas palabras. Detectamos también

que el término "(Aplausos)" podría ser también un buen delimitador. Sin embargo, se detectó que el uso de esta última palabra en ocasiones ocurre a la mitad de un discurso o intervención. Utilizando únicamente el primer término de corte, las doce mil páginas originales resultaron en 2,885 documentos independientes.

Table 1: Documentos por año

Diario de los debates	**Documentos generados**
1946	370
1952	428
1958	309
1964	736
1970	444
1976	598
Total	**2885**

2. Representar los subdocumentos por sus características semánticas, terminología utilizada, co-ocurrencia de términos y en general elementos que permitan distinguir los temas que se discuten.

   Para esta sección se utilizó el algoritmo doc2vec de la librería Gensim en Python. Uno de los elementos a considerar es que este algoritmo requiere de insertar cada uno de los documentos como un objeto LabeledSentence que requirió importantes transformaciones en los textos de entrada. Es indispensable cierto preprocesamiento en los datos: se probó el uso del algoritmo de Porter en español que se encuentra implementado en la librería nltk como SnowBall Stemmer; sin embargo, los resultados generaban dificultades de interpretación (palabras demasiado cortas, no se distinguían plurales y singulares o género masculino y femenino).

   El algoritmo devuelve una representación vectorial de las palabras del corpus con número de dimensiones que debe de ser indicado por el investigador. Es importante que el número de dimensiones sea grande (300-500); sin embargo, el tiempo de procesamiento crece conforme se incrementa el número de dimensiones de la representación. Se utilizaron para nuestro análisis 300 dimensiones, esto significa que cada palabra está representada por un vector numérico de 300 elementos. El algoritmo provee ciertas funciones que permiten medir la distancia semántica entre palabras utilizando la distancia coseno, donde se considera únicamente el ángulo que generan los vectores de dos palabras diferentes. Los resultados de este algoritmo basado en redes neuronales profundas permitieron comprobar su potencial para identificar similitudes entre palabras dados los contextos.

   Se analizaron algunos de estos resultados, por ejemplo, la palabra hombre tiene como elementos de mayor similitud las palabras "individuo",

"maestro", "pilar", "soldado", "viejo". La palabra mujeres dio como resultados "tradiciones", "minorías", "damas", "atenciones". Se realizaron varias pruebas, eliminando la puntuación, quitando las mayúsculas, removiendo los acentos, removiendo plurales y géneros. Al remover los signos de puntuación mejoró claramente el funcionamiento del algoritmo ya que la palabra "intereses", "intereses," e "intereses." se consideraban como diferentes. Probablemente no es necesario remover completamente la puntuación sino únicamente añadir espacios antes y después de los signos de puntuación, queda pendiente realizar este tipo de pruebas.

Se decidió no utilizar lematización de las palabras ya que se encontraron interesantes asociaciones que podrían perderse en el caso de ser aplicada. Por ejemplo, la palabra "hombre" se relaciona con "soldado" y "viejo" mientras que "hombres" se relaciona más con "campesinos" y "jovenes". Estos ligeros matices nos parecieron interesantes y por lo tanto decidimos conservarlos. El corpus resultante estuvo compuesto por 27 mil 480 palabras únicas. Algunos elementos interesantes que vale la pena recalcar vienen en la siguiente tabla:

Table 2: Similitudes entre palabras

Palabra objetivo	**1a similitud**	**2a similitud**	**3a similitud**
enero	octubre	abril	diciembre
izquierda	derecha	majestad	devolución
PRI	PAN	PARM	PNM
indigenas	pobres	necesitados	agricultores
cuba	suecia	panama	venezuela
el	un	este	ese
progreso	mejoramiento	engrandecimiento	bienestar
revolucion	democracia	dictadura	patria

Es notable cómo este algoritmo es capaz de detectar la semántica de ciertas palabras y su significado en el gobierno PRIista post-revolucionario.

Un elemento sumamente interesante del algoritmo doc2vec es que también genera una representación vectorial de cada documento. En nuestro caso cad,a documento se representa por un vector de 300 valores numéricos, por lo que es posible replicar el análisis a nivel de palabras entre los documentos para medir su similitud con base al ángulo generado por cada uno de los vectores en el espacio multidimensional.

Una vez obtenida la matriz que representa los documentos es posible pasar al siguiente paso de nuestra metodología. Se menciona entonces que se tiene una matriz de 2,885 renglones por 300 columnas.

3. Agrupar todos los documentos que por sus características morfológicas y semánticas se pueden considerar similares.

El siguiente paso fue la implementación de los algoritmos de aprendizaje no supervisado: K-Medias y Major-Clust.

Para el primero, se utilizó una distancia euclidiana entre los documentos, mientras que para el de Major-Clust se usó una distancia de similitud de cosenos. Es importante mencionar en este punto que la distancia por similitud de cosenos es un valor entre -1 y 1, donde los valores más cercanos a 1 representan una mayor similitud entre las palabras. Esto a diferencia de las llamadas distancias de Minkowski (que incluyen la euclidiana) en donde los elementos más cercanos tienen métricas más pequeñas. Esto es fundamental a la hora de implementar los algoritmos.

Para el algoritmo de K-Means se utilizaron las implementaciones de SciPy y de Scikit-Learn, se revisó que los resultados de ambas eran similares. Sin embargo el paquete de Scikit-Learn provee un mayor número de operaciones para la generación de múltiples modelos y una mayor facilidad de acceso a los resultados.

Uno de los retos que presenta el algoritmo de K-Medias es el de la determinación del número adecuado de clusters en los que se quieren clasificar los documentos. Para determinar este número se utilizó el criterio del codo, el cual compara las ganancias marginales de incluir un cluster más contra la suma de los errores cuadráticos promedio dentro de cada clúster. Según este criterio, el número de clusters que debería usarse es aquél en el que se marque un punto de inflexión en la gráfica. Para este trabajo se probó con valores desde 1 hasta 20 grupos. Se presenta a continuación una gráfica con el porcentaje de la varianza intra-cluster para cada valor seleccionado en el número de grupos. Por ejemplo, al elegir 5 grupos, se puede explicar el 36% de la varianza total en los datos. Aunque se detectaron varios puntos de inflexión en la gráfica (el criterio del codo indica que estos puntos de inflexión indican valores adecuados para el algoritmo) decidimos tomar en cuenta el factor de interpretabilidad de la información y consideramos que caracterizar 13 o 18 grupos sería complicado para este proyecto por lo que nos quedamos con 5 grupos. Esto claramente representa el riesgo de no detectar adecuadamente los grupos de temas de elecciones o presupuestales y queda pendiente en un futuro estudio revisar si un mayor número de grupos brindan mejores resultados.

Table 3: Documentos por cluster, K-Medias

Identificador del cluster	**Número de documentos**
Cluster 0	245
Cluster 1	1330
Cluster 2	374
Cluster 3	735
Cluster 4	201

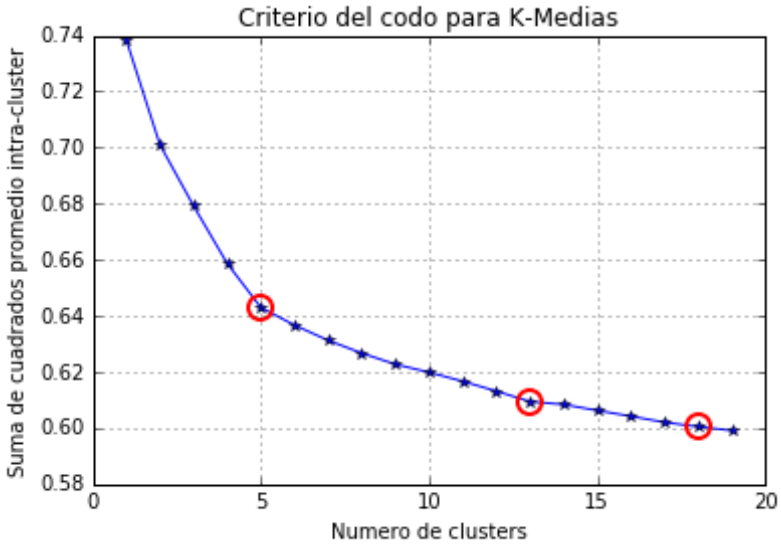

Figure 1: Selección de clusters

La distribución de los documentos originales por cada tipo de cluster después de utilizar K-Medias se muestra en la siguiente tabla:

El segundo algoritmo de agrupamiento que utilizamos fue el algoritmo de Major-Clust. Este algoritmo utiliza una lógica de agrupamiento por densidad basado en grafos. Este algoritmo es ampliamente utilizado para la caracterización de documentos, otra de sus bondades es que no es necesario indicar el número de grupos que se desea generar a priori, sino que el algoritmo termina cuando ningún nodo (documento) cambia su pertenencia de grupo. No existe aún una implementación del algoritmo en python por lo que fue necesario implementar el algoritmo basándonos en publicaciones que explican su funcionamiento por seudo-código.

MAJORCLUST.
Input. A graph $G = \langle V, E, \varphi \rangle$.
Output. A function $c : V \to \mathbf{N}$, which assigns a cluster number to each node.

(1) $n = 0, t = \mathit{false}$
(2) $\forall v \in V$ **do** $n = n + 1, c(v) = n$ **end**
(3) **while** $t = \mathit{false}$ **do**
(4) $t = \mathit{true}$
(5) $\forall v \in V$ **do**
(6) $c^* = i$ **if** $\left(\sum_{\substack{c(u)=i, \\ \{u,v\} \in E}} \varphi(u,v) \right)$ is max.
(7) **if** $c(v) \neq c^*$ **then** $c(v) = c^*, t = \mathit{false}$
(8) **end**
(9) **end**

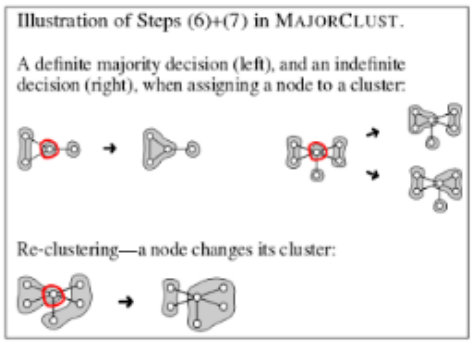

Figure 2: Selección de clusters

Los resultados obtenidos con el algoritmo de Major-Clust fueron inicialmente inadecuados, ya que al correr el algoritmo introduciendo los datos de manera ordenada se obtiene un solo cluster con más de 2,500 documentos y el resto clasificaba cada documento en su propio cluster. Después de una investigación sobre algoritmos de densidad se decidió que una manera de mejorar los resultados era alimentar las observaciones de manera aleatoria al algoritmo. Otro elemento que mejoró la diversificación de asignación a grupos fue la asignación aleatoria cuando un nodo pudiera ser asignado a varios grupos. Es importante recalcar que a pesar de estas mejoras, obtenemos de manera regular un cluster con un gran número de documentos y un número muy grande de clusters con cardinalidad de 1. Estamos conscientes que estos resultados no son adecuados y creemos que es necesario una mayor investigación con una colección de documentos clasificados por otro mecanismo para medir la precisión de este agrupamiento no-supervisado. Es posible que exista alguna deficiencia en nuestra implementación del algoritmo, aunque fue revisada de manera exhaustiva y no se encontró ningún error de conceptualización. Uno de los mejores resultados obtenidos, separaba los documentos en más de 1,000 grupos donde la moda de la cardinalidad era de 1. Decidimos tomar los 5 clusters que agrupan el mayor número de documentos aunque estamos conscientes de las grandes limitaciones de nuestros resultados. Se presentan a continuación el número de documentos que pertenecen a cada uno de los grupos.

Table 4: Documentos por cluster, MajorClust

Identificador del cluster	**Número de documentos**
Cluster 0	1692
Cluster 1	28
Cluster 2	20
Cluster 3	19
Cluster 4	18

4. Caracterizar los grupos de temáticas por términos multipalabra clave que permitan identificar las grandes temáticas que se discuten en la Cámara baja.

   Una vez que se tuvo lista el agrupamiento de los documentos de acuerdo a cada algoritmo de aprendizaje no supervisado se procedió a caracterizarlos. Para esto, se agruparon todos los documentos con la misma etiqueta de cluster en un solo documento. De esta forma, por ejemplo, resultaron cinco documentos grandes cuando se utilizó el algoritmo de K-Medias.

   Después de obtener estos documentos anidados por su clasificación, se utilizó el algoritmo de RAKE para extraer las tópicos clave multipalabra de cada uno. Aunque existe una implementación del método RAKE en el paquete nltk, éste no permite la modificación de parámetros importantes. Sin embargo, encontramos una implementación del algoritmo que sí permite la modificación de dichos parámetros, aunque éste no existe aún en un paquete auto-contenido de python y fue necesario realizar algunas adecuaciones. El código de la implementación se encuentra en las referencias de este documento y corresponde al repositorio de GitHub de Alyona Medelyan.

   Este algoritmo requiere de una lista de términos de paro en español -que fue descargada desde el paquete de lenguaje natural de Python-, el documento a caracterizar, así como los parámetros referentes a los caracteres que debe tener cada palabra (se eligió 5), la cantidad máxima de palabras que puede tener cada frase (también se escogieron 5) y la repetición mínima en la que la palabra deba estar en el texto (se eligió 20). Otro elemento importante a considerar es que este algoritmo requiere de los caracteres de puntuación y de los acentos originales del texto por lo que fue necesario reconstruir el texto original.

   Uno de los problemas que se presentó al implementar el algoritmo de RAKE fue que el tiempo de procesamiento era muy largo dado el volumen de los textos trabajados. Para solucionarlo, se decidió realizar muestras aleatorias de 40 documentos para cada cluster y posteriormente caracterizarlos. Esta solución parcial permitió que se pudiesen probar distintos parámetros hasta obtener los presentados en el presente documento. Cabe notar que los resultados de las tópicos latentes de los clusters con la selección aleatorizada de documentos fueron muy similares con respecto al

caso en el que se tomaban la totalidad de los mismos; lo que indica que las muestras utilizadas fueron representativas del corpus total.

Al revisar los resultados obtenidos con el algoritmo RAKE detectamos que existen varios términos multipalabra que se comparten entre grupos y aparecen de manera continua en todo el documento (como por ejemplo la palabra Aplausos). Al notar que estos resultados no permiten caracterizar de manera única cada uno de los grupos, tomamos la decisión de aplicar el algoritmo de extracción de términos multipalabra clave basados en corpus tf-idf.

5. Determinar cuáles de los términos multipalabra clave son únicos y permiten distinguir una temática de las demás.

   Como consecuencia del tamaño de los documentos que componen a cada cluster, la cantidad de palabras extraídas de cada grupo fue bastante alto. Para facilitar el análisis y marcar las diferencias entre los tópicos de cada cluster, se procedió a anidar las palabras clave de cada cluster y a realizar su representación vectorial a través de tf-idf. Fue posible realizar n-gramas de hasta 3 palabras, por lo que se identificaron aquellas frases que caracterizan a cada cluster, después de comparar con las palabras extraídas por los otros agrupamientos.

   La combinación entre los algoritmos de RAKE y el tf-idf es la mayor aportación metodológica que tiene este trabajo de investigación porque logra combinar las mejores cualidades de cada uno. Por un lado, RAKE utiliza la información contenida en cada cluster para extraer las frases clave; mientras que la representación vectorial de estas frases a través de tf-idf permite la comparación entre clusters a partir de un corpus mucho más reducido que el original, por lo que el costo computacional es mucho menor con respecto a aplicar este algoritmo sobre el corpus original. Este anidamiento, además, facilita el análisis de los tópicos característicos de cada cluster con respecto al caso en el que sólo se utiliza RAKE.

5 Evaluación

Se presentan a continuación los resultados utilizando los dos algoritmos de agrupamiento. Debido a las limitaciones discutidas por el algoritmo de Major Clust creemos que los resultados carecen de validez, pensamos que sería necesario tomar el cluster con mayor número de documentos y reinsertar al algoritmo de Major Clust para obtener mayores resultados. La realización de estas pruebas fueron imposibles por la limitante de tiempo y queda como un elemento adicional a explorar para futuras investigaciones.

Se intentó clasificar con una temática cada uno de los clusters encontrados en base al conocimiento previo que uno de los miembros del equipo tenía respecto a estos documentos y, aunque los resultados no fueron tan claros como se podría esperar, sí representan las principales temáticas que existen en los documentos.

A continuación se presentan los términos multipalabra clave encontrados con el algoritmo de K-Means:

	Cluster 0 - Presupuesto		Cluster 1 - Creación de Leyes		Cluster 2 - Miscelaneo		Cluster 3 - Electoral		Cluster 4 - Intervenciones Diputados	
Rango	termino	score	termino	score	termino	score	termino	score	termino	score
1	artículo	0.39	comisión	0.17	electoral	0.28	electoral	0.33	comisión	0.17
2	ley	0.14	ejecutivo	0.12	distrito	0.15	diputados	0.20	horas	0.14
3	comercio	0.10	votación	0.10	partido	0.13	distrito	0.17	artículo	0.10
4	federal	0.09	único	0.09	federal	0.12	diputados federales	0.15	año	0.10
5	comisión	0.08	aceptar	0.08	diputados	0.10	distrito electoral	0.13	grado	0.10
6	secretario	0.08	comisión permanente	0.08	distrito electoral	0.10	partido	0.13	secretario	0.08
7	nacional	0.08	decreto	0.08	pueblo	0.07	federales	0.11	usted	0.08
8	crédito	0.08	permanente	0.08	revolucionario	0.07	celebradas	0.09	distrito	0.08
9	público	0.08	proyecto	0.08	ley	0.06	federal electoral	0.09	república	0.08
10	hacienda	0.07	proyecto decreto	0.08	mayoría	0.06	mencionado	0.09	sesión	0.08
11	general	0.07	aceptar ciudadanía	0.07	colegio	0.05	comisión	0.09	lectura	0.07
12	iniciativa	0.07	aceptar ciudadanía mexicana	0.07	colegio electoral	0.05	federal	0.09	permanente	0.07
13	lectura	0.06	aceptar república	0.07	federal electoral	0.05	elecciones	0.08	conocimiento	0.07
14	secretaría	0.05	aceptar república presidente	0.07	institucional	0.05	propietario	0.08	designa	0.07
15	pública	0.05	acuerdo votación	0.07	popular	0.05	económica diputados	0.08	oficial	0.07
16	fracción	0.05	acuerdo votación nominal	0.07	comisión	0.05	económica diputados federales	0.08	general	0.06
17	acera	0.05	afirmativa mayor	0.07	cámara	0.05	ley	0.06	servicios	0.06
18	comisión nacional	0.05	afirmativa mayor asamblea	0.07	discusión	0.05	diputados federales celebradas	0.06	ustedes	0.06
19	egresos	0.05	anteriormente expuesto secretario	0.07	mexicana	0.05	federales celebradas	0.06	acuerdo	0.06
20	entrará	0.05	aprobado perder	0.07	nacional	0.05	mencionado distrito	0.06	ejecutivo	0.06

Figure 3: Resultados K-Medias

Y los resultados obtenidos con el algoritmo de Major Clust

	Cluster 0		Cluster 1 - Electoral		Cluster 2		Cluster 3		Cluster 4	
Rango	termino	score	termino	score	termino	score	termino	score	termino	score
1	comisión	0.12	diputados	0.15	artículo	0.16	impuesto	0.12	comisión	0.15
2	permanente	0.11	comisión	0.11	nacional	0.14	diputados	0.10	infancia	0.10
3	permiso	0.10	electoral	0.10	partido	0.09	ley	0.10	instituto	0.10
4	corte	0.08	aprueba	0.09	aérea	0.09	dictamen fecha	0.07	cámara	0.10
5	nacional	0.08	artículo	0.09	diputados	0.08	dictamen fecha presidente	0.07	secretaría	0.10
6	suprema	0.08	cámara	0.09	federal	0.08	fecha presidente	0.07	gobernación	0.09
7	suprema corte	0.08	federal electoral	0.07	secretario	0.08	ingreso	0.07	comisión puntos	0.08
8	congreso	0.08	partido	0.07	distrito	0.08	utilidades	0.07	mayor	0.08
9	votación	0.08	discusión	0.06	comisión	0.07	artículo	0.07	aprueba	0.07
10	comisión permanente	0.07	federal	0.06	secretaría	0.07	fracción	0.07	diputados	0.07
11	trabajadores	0.07	presidente	0.06	albarrán	0.06	presidente	0.07	comisión permanente	0.07
12	condecoración votación	0.06	propietario	0.06	asociaciones	0.06	secretaría	0.07	permanente	0.07
13	confirió	0.06	votación	0.06	diputados secretario	0.06	votación	0.07	puntos	0.07
14	consideración gobernación	0.06	federales	0.06	fernández	0.06	diario	0.06	licenciado	0.06
15	exteriores comisión	0.06	ley	0.06	fernández albarrán	0.06	causantes	0.06	76	0.05
16	gobernador trabajadores	0.06	primera	0.06	fuerza	0.06	consideración comisión	0.06	76 constitucional	0.05
17	relaciones exteriores comisión	0.06	constitución	0.05	fuerza aérea	0.06	aprueba	0.05	76 constitucional secretaría	0.05
18	secretaría comisión	0.06	distrito	0.05	instituciones	0.06	cámara	0.05	abre efectos	0.05
19	artículo	0.06	siguientes	0.05	lima	0.06	discusión	0.05	abre efectos secretarios	0.05
20	cámara	0.06	5o artículo anteri	0.04	lima josé	0.06	méxico	0.05	aceptar acta	0.05

Figure 4: Resultados Major Clust

Es evidente que debido a la poca representatividad de cada uno de los clusters encontrados por Major Clust iba a ser complicado encontrar la temática correspondiente. En efecto, los resultados de Major Clust no pueden ser utilizados en su estado actual. Proponemos el reinsertar el cluster con el mayor número de documentos en otro grafo y volver a correr el algoritmo con la esperanza de obtener mejores resultados.

Por otro lado el análisis con el algoritmo de K-Means arrojó resultados interesantes que permiten distinguir las siguientes temáticas:

1. Temas de presupuesto
2. Temas Legislativos
3. Misceláneos (no fue posible determinar una temática guia)
4. Temas electorales

5. Intervenciones de diputados o discursos

Es necesario realizar pruebas de caracterización con un mayor número de grupos en el caso de K-Means para revisar que se están encontrando todas las temáticas que realmente se discutieron en la Cámara.

6 Conclusiones

La metodología que proponemos en este estudio puede utilizarse para cualquier tipo de documentos donde se toquen una diversidad de temas y que se publican de manera regular. Por ejemplo, otra serie de documentos en los que se podría utilizar nuestra metodología son los informes de gobierno presidenciales. En general, esta metodología se puede aplicar a cualquier evento recurrente donde se repitan una serie de temas diversos, como por ejemplo juntas de oficina, asambleas de condóminos y, en general, cualquier evento que genere transcripciones textuales.

Los resultados obtenidos del análisis de los Diarios de Debates distingue diferentes temáticas; sin embargo, no creemos que se puedan utilizar en su estado actual para realizar clasificación. Es necesario extender la presente investigación con una visión de medición de la precisión de clasificación. El siguiente paso es tomar una selección de al menos 120 subdocumentos y revisarlos manualmente para ser etiquetados. Esta etiquetas permitirán revisar de manera más precisa que las temáticas que se están detectando por esta metodología son adecuadas y que los grupos en los que se subdivide son suficientes.

La originalidad de nuestro trabajo reside en el uso de algoritmos de análisis de texto muy poderosos que combinados permiten generar un valor agregado importante con muy poco conocimiento previo de su contenido. Algunos de los elementos que son posibles extensiones del presente documento se listan a continuación:

- Automatizar la fase de separación de las transcripciones en bloques temáticos.
- Analizar la sensibilidad de nuestra metodología al incrementar o reducir el número de vectores de la representación vectorial. Realizar la caracterización de documentos con un mayor número de clusters usando el algoritmo de K-Means.
- Mejorar el algoritmo de Major Clust para poder anidar los resultados en sus diferentes iteraciones.
- Probar otros algoritmos de agrupamiento como por ejemplo el uso de técnicas de clustering espectral.
- Aplicar el algoritmo de TF-IDF a la totalidad de los documentos considerando que cada grupo representa un documento y el corpus es la totalidad de los textos.

- Comparar los resultados del algoritmo RAKE combinado con TF-IDF, TF-IDF solo y una propuesta de aplicar TF-IDF y después aplicar RAKE sobre los resultados de los términos multipalabra.
- Realizar un esfuerzo de etiquetado manual para determinar la precisión de nuestro análisis.

7 Referencias

1. Berry, M, Kogan, J. (2010). *Text Mining, Application and Theory.* Wiley.
2. Le, Q, Mikolov, Tomas. (2014). *Distributed Representations of Sentences and Documents.*
3. Stein, B, Meyer zu Eissen, M. (2002). *Document Categorization with MAJORCLUST.*
4. Medelyan, A. (2015). *NLP keyword extraction tutorial with RAKE.*
5. Medelyan, A. (2015). *RAKE Implementation.*
   Fuente: https://github.com/zelandiya/RAKE-tutorial