Métodos Analíticos para Texto: Identificación, Clasificación y Agrupamiento de Tópicos de Tuits

David Ricardo Jaramillo
Maestría en Ciencias en Computación
83915

Ángel Farid Fajardo
Maestría en Ciencias en Computación
153464

Jorge Adrián Sánchez
Maestría en Ciencias de Datos
116369

Omar Díaz Landa
Maestría en Ciencias de Datos
114041

RESUMEN

En este trabajo se combinaron algunos algoritmos y modelos estadísticos utilizados en el Procesamiento de Lenguaje Natural, con el fin de recuperar información y agrupar por tópicos un conjunto de documentos recolectados de la red social de Twitter. La recolección de textos para este trabajo es a partir del diseño e implementación de una arquitectura de producto de datos, la cual permite recabar una gran cantidad de mensajes los cuales forman el corpus que analizamos. [1]

1. INTRODUCCIÓN

Con el ímpetu en que las nuevas Tecnologías de Información y Comunicación han afectado el acceso a la información y a la promoción democrática de los ciudadanos, las redes sociales se han convertido en un nuevo canal para influir en la toma de decisiones de políticas públicas. Es por esta razón que es importante el desarrollo de herramientas que permitan visualizar información relevante y actualizada a los ciudadanos, y de esta manera proveer un mejor contexto sobre aquellos temas que afectan a la sociedad, como las elecciones presidenciales de una nación.

Por tal motivo, hemos desarrollado una metodología que permite encontrar los tópicos más relevantes en conversaciones referentes a la candidatura de Donald Trump a la presidencia de Estados Unidos, los cuales se basan en modelos estadísticos aplicados en el Procesamiento de Lenguaje Natural (PLN), como los sistemas de vectorización *tf-idf*, los modelos de análisis semántico y la Descomposición de Valores Singulares bajo un modelo distribucional.

Debido los 65 millones de usarios que la red social Twitter tiene en EE.UU., este análisis ofrece una buena aproximación sobre aquellos tópicos frecuentemente mencionados en relación a la candidatura de @realDonaldTrump [15].

[1]Este proyecto de investigación funge como el resultado de combinar los proyectos finales de la materia de Arquitectura de Producto de Datos impartida por el Dr. Adolfo de Unánue y la materia de Métodos Analíticos para Texto impartida por el Mtro. Victor Mijangos, de la Maestría en Ciencia de Datos.

2. ESTADO DEL ARTE

Dentro del análisis semántico del PLN, se han implementado distintos algoritmos que buscan extraer el significado de técnicas, como el Análisis Semántico Latente [6] [7]. Comunmente, este tipo de algoritmos parten de una hipótesis distribucional, la cual establece que las palabras que ocurren en contextos similares, conservan significados similares. De tal manera es posible obtener una relación de términos (sinónimos y polisemia) o inclusive agrupar términos desde una perspectiva de contexto-significado.

En las siguientes sub-secciones se expondrán de manera breve una descripción de los conceptos y algoritmos que se utilizarón en este proyecto, para la identificación, clasificación y agrupamiento de tópicos.

2.1 Leyes empíricas del lenguaje

Exiten algunas leyes empíricas del lenguaje la cuales dan una explicación acerca del comportamiento de las palabras dentro del contexto de PLN, como la ley de Zipf, la fórmula de Mandelbrot y la ley de Herdan, que son descritas por las ecuaciones 1, 2 y 3 respectivamente, donde f es la frecuencia de una palabra en un corpus, r es el rango de una palabra, $alpha$ es una constante, P y p son parámetros del texto, N es el tamaño del corpus (número de tokens) y t es en número de tipos en el texto.

$$f \propto \frac{1}{r^a} \tag{1}$$

$$f \propto \frac{P}{(r+\rho)^{-\alpha}} \tag{2}$$

$$t \propto N^{\alpha^{-1}} \tag{3}$$

Con base a estas leyes empíricas es posible generar modelos estadísticos que sean capaces de caracterizar y clasificar un conjunto de palabras. Por ejemplo, la ley de Zipf ayuda a entender el por qué los valores que se generan a partir de *tf-idf* deben de ser ponderados por la frecuencia en que las palabras aparecen en un corpus, deduciendo que las palabras "funcionales" son muy frecuentes y poseen un valor semántico significativo.

2.2 Vectorización TF-IDF

La vectorización de palabras se basa en los valores estadístico obtenidos a partir de tf-idf *(term frequency inverse-document frequency)* [12], que es una representación ponderada de vectores del modelo de Bolsa de palabras *(bag-of-words)* utilizada en PLN, en el cual el subconjunto de palabras de un corpus es representado en un espacio vectorial, cuya dimensión depende del número de documentos analizados y de la cantidad de palabras diferentes del corpus [5].

El objetivo de *tf-idf* es capturar la frencuencia relativa de las palabras por documentos, ponderando dicho valor de acuerdo a la frecuencia máxima que tiene una palabra dentro del mismo documento. Este ajuste sirve para crear una proporción entre palabras comunes *(stop-words)*, como los artículos o las preposiciones que generalmente tienden a aparecer en mayor frecuencia [11], y las palabras semánticamente relevantes.

El *tf-idf* es calculado con base a las siguientes ecuaciones:

$$tf(w) = c + (1-c)\frac{C(w:d)}{max_i\{C(w_i:d)\}} \tag{4}$$

$$idf(w) = log\left(\frac{\|\mathbb{C}\|}{\|\{d_i : w \in d_i\}\|}\right) \tag{5}$$

$$A = a_{i,j} = \begin{cases} 0 & \text{si } w_i \notin d_j \\ tf(w_i) * idf(w_i) & \text{si } w_i \in d_j \end{cases} \tag{6}$$

donde $tf(w)$ representa la frecuencia ponderada de un término en un documento, e $idf(w)$ representa a la frecuencia inversa de un término en los documentos de un corpus. El término c es una constante arbitraria tal que $0 <= c <= 1$, C representa el conteo de las palabras, $||C||$ representa la cardinalidad del corpus y d_i representa el documento en donde la palabra aparece. Los elementos de la matriz A son calculados a partir de la multiplicación del *tf y idf* por palabra, como lo indica la ecuación 3.

De manera intuitiva se aprecia que el valor *tf-idf* de una palabra en un corpus es directamente proporcional a la frecuencia relativa de las palabras en un docuemtno específico, e inversamente proporcional a la frecuencia de esa misma palabra entre los documentos del corpus. De esta manera se le otorga un peso relativo a las palabras de un documento, el cual es afectado si esa palabra aparece en todos los documentos, debido a que aquellas palabras comúnes no proporcionan gran significado semántico de interés, como son los artículos o preposiciones [10].

2.3 Modelo Distribucional Semántico Simple

La hipótesis detrás del modelo distribucional semántico simple (MDSS) establece que las palabras que ocurren en contextos similares son semánticamente similares. De esta manera, se establece que en un MDSS, cada palabra es representada por un vector, en donde los valores de cada componente del vector se calculan en función del número de veces en que las palabras ocurren en determinados contextos linguísticos, permitiendo aproximar un significado semÃątico por palabra de acuerdo a una cierta distancia entre vectores.

La principal caracerística de este modelo es la capacidad de determinar cierta similitud o semejanza entre palabras, en base a los contextos en que aparecen esas palabras.[2] Teniendo en cuenta esto, es posible generar un MDSS que determine la relación semántica entre un conjunto de palabras.

Un MDSS genera matriz A de $w_i \times d_n$, donde w reprensenta a las palabras del corpus y d el número de documentos, tal que:

$$A = a_{i,j} = \begin{cases} C(w_{i,j}) & \text{si } w_{i,j} \in d_n \\ 0 & \text{si } w_{i,j} \notin d_n \end{cases} \tag{7}$$

Los vectores distribucionales son obtenidos a partir de la frecuencia de una palabra dado un contexto (o un documento), de esta manera se define una matriz de palabras-documentos. De tal forma, es posible proveer de una cuantificación precisa de similitud semántica, derivada de la representación geométrica de las palabras-contextos, a partir de una medida de distancia entre las palabras. Es comunmente utilizada la distancia coseno entre palabras bajo un cierto espacio vectorial, de tal manera que los ángulos de los vectores-palabra obtenidos definen la similitud semántica entre palabras [4].

2.4 Descomposición en Valores Singulares

La Descomposición de Valores Singulares (DVS) es una herramienta que se basa en descomponer una matriz en base

[2]Una de las decisiones más importantes para la implementación de los modelos MDSS es la defínción del *contexto* para el conteo de ocurrencias. Distintas definiciones de contexto tienden a capturar distintas tipos de similutud semántica o de relación. Para propósitos de esta investigación el *contexto* se tomará a partir de documentos para capturar de mejor manera las relaciones por tópico.

en sus *eigenvalores* y sus *eigenvectores*, de tal forma que se puede aproximar una matriz de datos a partir de los valores o características que representan la mayor varianza, evitando trabajar con valores que no se consideran estadísticamente independietes; por tal motivo, esta herramienta algebraica se utiliza para trabajar en espacios de menor dimensión, en este tipo de análisis de datos.

Al reducir la dimensionalidad de un set de datos y ordernarlos de manera descendente de acuerdo a su varianza, nos permite ignorar la variación proveniente de datos dependientes de otras variables, a partir de cierto límite seleccionado de manera arbitraría, para reducir masivamente el número de dimensiones-datos sin perder la relación estructural original [1].

La DVS se basa en el teorema de algebra lineal que establece que una matriz A se puede descomponer en el producto de tres matrices - una matriz ortonogal U, una matriz diagonal S y su matriz ortogonal transpuesta V:

$$\mathbf{A}_{m,n} = \mathbf{U}_{m,m}\mathbf{S}_{m,n} \times \mathbf{V}^{*}_{n,n} \tag{8}$$

En donde los eigenvalores y eigenvectores pueden ser encontrados a partir del trato que se le puede dar tanto a la matriz U_{mm} o la matriz $V^{\intercal}_{nn}$, como un sistema de ecuaciones lineales y resolviendo por los valores de las variables que conforman los componentes del eigenvector.[3]

Para el análisis semántico, la DVS implica que a partir de los datos orginales, que usualmente consiste en una matriz de palabras $\times$ documentos, su descomposición arroja componentes linearmente independientes de palabras. De alguna manera, estos componentes son una abstracción de las correlaciones ruidosas que se encuentran en los datos originales, dejando valores que mejor aproximan la estructura interna de los datos a través de cada dimenisión independientemente. Adicionalmente, la DVS permite visualizar a las palabras que comparten un significado similar (semánticamente hablando), ya que tendrán una menor distancia en un espacio vectorial, mientras que las palabras que son semánticamente distintas conservan una distancia mayor.

Cabe destacar que la DVS es la base del Análisis Semántico Latente, el cual es una metodología de indexación y recuperación de información. Este enfoque establece una estructura implícita de los términos en documentos con base en la búsqueda de palabras clave ademÃąs de indexar documentos en base a las palabras clave las cuales son utilizadas como consulta (*queries*). Estas consultas son efectuadas a partir de la ortogonalidad que establecen las matrices de documentos-palabra, y la distancia coseno entre los vectores palabra representados en un espacio vectorial de menor dimensión mediante DVS [9] [3].

2.5 Agrupamiento de textos por tópicos

[3]Para ver un ejemplo de este modelo visite el documento titulado *Singular Value Decomposition Tutorial* de Kirk Baker encontrado en la sección de referencias de este documento.

Para agrupar palabras o textos no estructurados por tema o significado, es común aplicar algoritmos como K-medias [8], *Spectral clustering* [16] o *MajorClust* [13] sobre grandes una gran cantidad de datos, pues es dificil contar con un corpus suficiente y debidamente etiquetado por la gran cantidad de n-gramas [17] que se pueden obtener de fuentes de datos no estructurados como lo es twitter.

Para poder aplicar estos algoritmos, es necesario transformar el corpus a un espacio vectorial multidimensional que caracterice las palabras y n-gramas de acuerdo a su valor semÃąntico en el corpus.

La expresión vectorial de los n-gramas en grandes cantidades de datos suele tener muchas dimensiones. Para lidiar con ello, se aplican métodos de reducciÃşn de dimensionalidad como SVD [2].

3. METODOLOGÍA

El objetivo de este proyecto es la aplicación de algoritmos y modelos estadísticos que permitan identificar y agrupar por tópicos a los documentos de un corpus compuesto por tuits. A continuación se describirá la metodología utilizada para esta investigación en dos subsecciones: (3.1) Arquitectura de Producto de Datos y (3.2) Identificación, Clasificación y Agrupamiento de Tópicos.

3.1 Arquitectura de Producto de Datos

En este proyecto se ha construído una herramienta de *streaming* que recaba automáticamente tuits, los cuales son objetivos a la candidatura de Donald Trump a la presidencia de Estados Unidos. Nuestra arquitectura es capazar de recibir los tuits que contienen una mención al usuario de @realDonaldTrump, el hasthag #Trump o el hastahg #trumo dentro de la cadena del tuit.

Esta arquitectura de producto de datos recaba la estructura de los tuits y los organiza en un archivo .JSON al que se añaden, conforme se tuitean, a un corpus compuesto de documentos y separado por pipes. Además, se ha diseñado un monitor para recabar el numero total de tuits y retuits, el top-rank de users que escriben los tuits y el top-rank de hashtags como lo muestra la Figura 1.

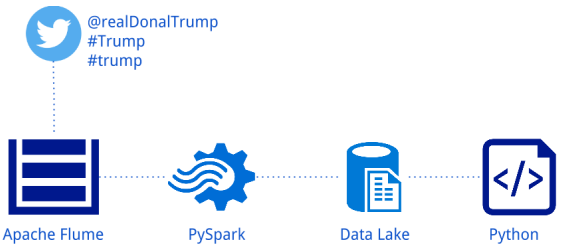

Figura 1: Diagrama de arquitectura de producto de datos, que refleja el mecanismo de recolección de tuits implementado

Como resultado de esta implementación, obtuvimos un cor-

pus conformado por aproximadamente 250,000 tuits (documentos), donde cada tuit tiene aproximadamente 140 caracteres.

3.2 Identificación Clasificación y Agrupamiento de Tópicos

Debido a la recolección exhaustiva de documentos que realizamos, nuestro corpus conserva una representatividad robusta sobre los temas que nos interesan. De manera que las aproximaciones estadísticas son indispensables al analizar la relación que existe entre palabras (términos) y documentos. El modelo aplicado utiliza métodos estadísticos, los cuales son normalmente empleados en PLN, como lo es la extracción de términos relevantes, la reducción de dimensionalidad mediante SVD (Figura 2), y clasificación de términos-documento mediante algoritmos de agrupamiento.

Figura 2: Diagrama del proceso para un análisis semántico lantente [14] sobre los textos recolectados de Twitter.

En términos generales, el resultado de esta propuesta es inducir el significado de palabras basándose en el contexto que tienen los términos dentro de un tuit (es decir, un tuit es considerado como el contexto de las palabras que lo componen).

De manera sistemática, el corpus conformado por tuits será representado en un espacio vectorial multidimensional, permitiendo de esta manera aplicar métodos estadísticos a estos datos. Esta representación vectorial del corpus analizado se basa en el modelo de bolsa de palabras *tf-idf*. A la matriz resultante de *documentos* $\times$ *palabras*, se le aplicará una función de composición por suma, para generar una estructura de datos de diccionario, en donde cada palabra tendrá asociado un peso que representa la importancia del término en el corpus. Una vez que tenemos este diccionario de palabras y pesos, se le aplicará un algoritmo de ordenamiento (función *sort()* de Python®), para detectar los prinicipales tópicos-palabras que se relacionan con la candidatura de Donald Trump a la presidencia de Estados Unidos.

Para disminuir la complejidad del algoritmo de agrupamiento que aplicamos (*K-medias*), aprendizaje no supervisado), es necesario reducir la dimensionalidad de nuestro corpus representado en un espacio vectorial dado, por lo que se aplica DVS. Apartir de esta descomposición, se elige de manera arbitraria x's atributos que representen de manera basta a las palabras del documento, ya que se eligen las variables que aumentan la varianza de los datos.

Una vez que tenemos una representación del corpus con una baja dimensionalidad, se pueden extraer tópicos relevantes, agrupar palabras por significado semántico, o inclusive relacionar tópicos con ciertas estructuras como los hashtags. Para tal fin, se tomará a la distancia coseno (ecuación 9, donde A y B son la representación de dos palabras en un espacio vectorial) para determinar la similitud entre palabras, tópicos o hashtags.

$$cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \sum \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{9}$$

4. IMPLEMENTACIÓN

El corpus utilizado para la experimentación corresponde a una muestra de aproximadamente 2000 tuits de dos fechas distintas.

El resultado de aplicar *tf-idf* a este corpus formando n-gramas de 2 a 4 palabras da como resultado una matriz de 55802 términos x 1865 documentos.

Para hacer más simple el análisis y facilitar la computabilidad de los datos, se utilizó la composición por suma y ordenamiento de los vectores de términos para escoger los 100 n-gramas más importantes obteniendo como resultado una matriz de 100 términos x 1865 documentos.

Posterior a esto, se aplicó SVD para reducir las dimensiones aún más de forma que elegimos conservar 2 características de los n-gramas, con la finalidad de obtener una representación gráfica que permita un análisis visual de los resultados.

Posteriormente se agruparon los n-gramas mediante el algoritmo de k-medias lo cual permite realizar un análisis, a nivel conversación, sobre los conjuntos formados por el algoritmo.

Se probaron distintos tamaños de n-grama, obteniendo resultados más interpretables con formaciones de 2 a 4 palabras.

Experimentando con el diccionario de términos completo, se observó que los términos se aglomeraban en un solo clúster. Analizamos los n-gramas detectando que el término *trump* se repetía en casi todas las composiciones. Por ello, se eliminó dicho término del diccionario con lo cual, observamos la formación de n-gramas más interpretables y clústers mejor distribuidos.

5. RESULTADOS Y EVALUACIÓN

A continuación se presenta el resultado de aplicar la experimentación descrita en la sección anterior:

Cluster 0 : [u'#trump2016 #trumptrain', u'@cenkuygur young', u'@cenkuygur young turks', u'@georgelopez thank destroying', u'camps offered', u'camps offered host', u'camps offered host cross', u'cross partisan', u'destroying young', u'destroying young people', u'destroying young people s', u'host cross', u'host cross partisan', u'host cross partisan debate', u'minds proud', u'minds proud rt', u'minds proud rt @princesslylia_', u'mr trump', u'offered host', u'offered host cross', u'offered host cross partisan', u'partisan debate', u'people s', u'people s minds', u'people s minds proud', u'proud rt', u'proud rt @princesslylia_', u'proud rt @princesslylia_ @georgelopez', u'reached @berniesanders', u'reached @berniesanders camps', u'reached @berniesanders camps offered', u'rt @cenkuygur', u'rt @cenkuygur young', u'rt @cenkuygur young turks', u'rt @danscavino', u'rt @elizabethforma', u'rt @foxnews', u'rt @georgelopez', u'rt @georgelopez thank', u'rt @georgelopez thank destroying', u'rt @princesslylia_', u'rt @princesslylia_ @georgelopez', u's minds', u's minds proud', u's minds proud rt', u'thank destroying', u'thank destroying young', u'thank destroying young people', u'turks reached', u'turks reached @berniesanders', u'turks reached @berniesanders camps', u'wall support', u'young people', u'young people s', u'young people s minds', u'young turks', u'young turks reached', u'young turks reached @berniesanders']
Cluster 1 : [u'rt @always_trump']
Cluster 2 : [u'donald trump']
Cluster 3 : [u'united states']
Cluster 4 : [u'tax returns']
Cluster 5 : [u'white house']
Cluster 6 : [u'trump s']
Cluster 7 : [u'rt @loudobbs']
Cluster 8 : [u'america great', u'make america great']
Cluster 9 : [u'stranded marines']
Cluster 10 : [u'make america']
Cluster 11 : [u'@cenkuygur young turks reached', u'@georgelopez thank', u'@georgelopez thank destroying young', u'@jonnot literally', u'@jonnot literally love', u'@jonnot literally love cnn', u'@princesslylia_ @georgelopez', u'cnn didn', u'cnn didn cut', u'cnn didn cut reply', u'cut reply', u'cut reply tweet', u'cut reply tweet screenshot', u'didn cut', u'didn cut reply', u'didn cut reply tweet', u'literally love', u'literally love cnn', u'literally love cnn didn', u'love cnn', u'love cnn didn', u'love cnn didn cut', u'reply tweet', u'reply tweet screenshot', u'reply tweet screenshot dm0dgjcra2', u'rt @jonnot', u'rt @jonnot literally', u'rt @jonnot literally love', u'screenshot dm0dgjcra2', u'tweet screenshot', u'tweet screenshot dm0dgjcra2']

El primer cluster muestra una serie de palabras claves, interesantes que a primera vista no parece tener sentido alguno. Sin embargo, haciendo una búsqueda de tendencias en internet sobre estos términos se pude demostrar que nuestro algoritmo es capaz de encontrar no solo tópicos importantes, sino conversaciones.

El cluster 0, hace referencia a una de las conversaciones más importantes que se dieron durante la captura de nuestros tuits. El día 25 de mayo a través de twitter Cenk Uygur, posteó que a través del programa online que dirije, titulado *The Young Turks (TYT)*, se llevaría a cabo un debate entre Donald Trump y su rival y opuesto de extremo liberal, Bernie Sanders. Además, ese mismo día por la noche Donald Trump, en su participación en el programa *Jimmy Kimmel Live!* aceptó la propuesta de entablar en un debate con Bernie Sanders. Como es de esperarse, desde ese mismo día, toda la conversación se volcó sobre este tema como lo capturan nuestros resultados.

De la misma, manera el cluster 11 capturó la converación que se dio con uno de los blogeros más importantes de estados unidos Jonno Turner, que ese mismo día posteo un tweet que tuvo un alto impacto en los comentarios hacia @readlDonaldTrump: *I literally love it that CNN didn't cut the reply to this @realDonaldTrump tweet from the screenshot* Este post tuvo, una influencia de grado 1 de 20 retweets y 30 mil likes.

Por otro lado, es importante destacar que los otros clusters capturan los tópicos clave que ha mantenido Trump en su carrera política. El término más utilizado dentro de los tuits corresponde a *https* 3 que nos indica que son enlaces a paginas web. En los terminos mayormente mencionados, hacen referencia a su contricantes Hillary Clinton y Bernie Sanders.

Como lo muestra la Figura 3, la representación espacial de vectores nos indica cierta clusterización acerca de ciertas

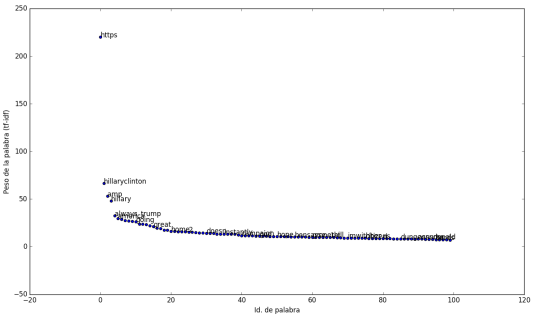

Figura 3: La ley de Zipf modela la importancia de una palabra, con respecto a su contexto

palabras y algunos datos atípicos.

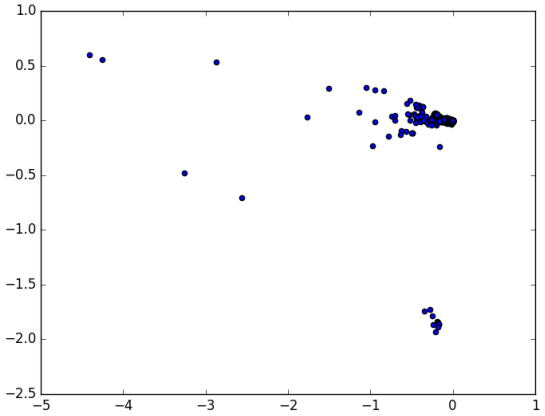

Figura 4: Representación de palabras "top" en un espacio vectorial de dos dimensiones

6. CONCLUSIONES

El modelo LSA (Latent Semantic Analysis) en conjunto con un algoritmo de agrupamiento *K-medias* permite realizar una análisis político-social, a partir de los datos provenientes de una red social como Twitter.

Esta implemetación es capaz de identificar a aquellos usuasrios con los que hay mayor interacción en relación a las temáticas obtenidas, así como los *hashtags* que las caracterizan.

El análisis de los *clusters* obtenidos de esta metodología revela los principales temas tratados al entorno de Donald Trump. Así mismo, observamos que aquellos *clusters* con mayor cantidad de términos caracterizan a aquellos sucesos importantes de mayor impacto en un periodo de tiempo.

Para una interpretación correcta y completa con base en los tuits, es necesaria una recolección y análisis de tuis en diferentes periodos de tiempo, de tal manera que se obtengan temáticas y conversaciones diversas en tornno a un tema central, que en esta caso es Donald Trump y las personas

más cercanas a él.

Para analizar documentos de texto relacionados con Twitter, es de suma importancia el desarrollo de un sistema que permita recolectar datos de manera masiva con el fin de generar un corpus de trabajo que sea robusto y representativo.

Como trabajo futuro, sería interesante comparar esta medología contra otras formas de vectorización y agrupamiento, como lo son *word embeddings, major clust* entre otros. Así mismo, aplicar estas nuevas metodologías a diferentes contextos de información.

7. REFERENCIAS

[1] K. Baker. Singular Value Decomposition Tutorial. *Ohio State University*, https://goo.gl/kqSs8d (accessado el 16 de mayo de 2016), 2013.

[2] L. Cagnina, M. Errecalde, and P. Rosso. Algoritmos bio-inspirados aplicados a tareas de clasificación de textos cortos. *IV Jornadas TIMM Tratamiento de la Información Multilingüe y Multimodal 7 y 8 de abril de 2011*, page 17.

[3] S. T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188âĂŞ230, 2004.

[4] M. B. et. al. Frege in Space: A Program for Compositional Distributional Semantics. *Linguistic Issues in Language Technlogy -LiLT*, 9(6):241–346, 2014.

[5] W. S. et. al. A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.

[6] T. Hofmann. Probabilistic latent semantic analysis. *In Proc. of Uncertainty in Artificial Intelligence, UAI 99*, pages 289–296, 1999.

[7] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, (42):177–196, 2001.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[9] C. D. Manning. Introduction to Information Retrieval. *Cambridge University Press*, http://nlp.stanford.edu/IR-book/ (accessado el 16 de mayo de 2016), 2008.

[10] J. Ramos. Using TF-IDF to Determine Word Relevance in Document Queries. *Rutgers University*, https://goo.gl/wQV7Z7 (accessado el 16 de mayo de 2016)., 2003.

[11] S. Robertson. Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60, 2004.

[12] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[13] B. Stein and S. M. Eißen. Document categorization with majorclust. In *Proc. 12th Workshop on Information Technology and Systems*. Citeseer, 2002.

[14] A. Thomo. Latent semantic analysis tutorial. *Victoria, Canda.[online] Available at:.[29 July 2012]*, 2009.

[15] Twitter. Company. *Twitter Usage/Company Facts*, https://about.twitter.com/company (accessado el 16 de mayo de 2016)., 2016.

[16] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[17] X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 697–702. IEEE, 2007.