**PROYECTO INTERCICLO IA+SE**

**Universidad Politécnica Salesiana**

**Narcisa Araujo, Roberto Pacho**

[**naraujop@est.ups.edu.ec**](mailto:naraujop@est.ups.edu.ec)

**RESUMEN:**

Los métodos de búsqueda son muy importantes en nuestro mundo actual ya que ellos nos permiten en base a ciertos criterios poder recomendar soluciones acordes a las necesidades del usuario mediante el desarrollo del presente documento presentaremos algunos problemas y soluciones que nos hemos planteado para dar soluciones optimas en situaciones de la vida real.

**PALABRAS CLAVE:** Inteligencia Artificial, Neo4j, Java, Algoritmos, Grafos, Python, etc.

**ABSTRACT:**

Search methods are very important in our world, since they allow us, based on certain criteria, to be able to recommend solutions according to the user's needs. Through the development of this document, we will present some problems and solutions that we have set ourselves to give optimal solutions in real life situations.

**KEY WORDS:** Artificial Intelligence, Neo4j, Java, Algorithms, Graphs, Python, etc...

1. **INTRODUCCION**

Para el desarrollo de nuestro sistema de búsqueda de datos de candidatos en twitter desde la base que es el conocimiento de cada uno de los algoritmos que previamente seleccionamos a continuación se detalla de manera general cada uno de ellos a fin de poder de una mejor manera el presente trabajo.

Al momento de nuestra elección sobre que algoritmos vamos a utilizar depende en gran medida del caso de uso que le vayamos a dar, para ello puede influir elementos como:

1. **“ALGORTIMOS DE DETECCION DE COMUNIDADES”**

**LOUVAIN**

Los algoritmos de detección de comunidades cumplen con una función de gran importancia para el análisis de datos en grafos que contienen información compleja, Estos tomando en cuenta sus características, formas de relación y diferentes tipos de conexiones nos ayudan a establecer o determinar la existencia de comunidades complejas de datos.

Dentro de este conjunto importante de algoritmos podemos contar el algoritmo de Louvain, el algoritmo de propagación de etiquetas y dos tipos interesantes de algoritmos orientados a estudiar diferentes tipos de grafos. Para entender el funcionamiento de grafos dirigidos contamos con el algoritmo de componentes fuertemente conectados y para grafos no dirigidos el algoritmo de componentes débilmente conectados.

1. **“ALGORTIMOS DE SIMILITUD”**

**SIMILITUD DE NODO**

Los algoritmos de similitud emplean comparaciones de conjuntos para puntuar qué tan parecidos son los nodos individuales en función de sus vecinos o propiedades. Las propiedades y atributos de los nodos se utilizan para puntuar la semejanza entre los nodos. Este enfoque se utiliza en aplicaciones como recomendaciones personalizadas y desarrollo de jerarquías categóricas.

Dentro de este conjunto de algoritmos contamos con uno de los algoritmos el algoritmo de similitud de nodos, este algoritmo compara un conjunto de nodos en función de los nodos a los que están conectados. Dos nodos se consideran similares si comparten muchos de los mismos vecinos. Node Similarity calcula las similitudes por pares según la métrica Jaccard, también conocida como Jaccard Similarity Score.

1. **“LOS ALGORTIMOS DE CENTRALIDAD”**

**CERCANIA CENTRALIDAD**

El algoritmo calcula la suma de sus distancias a todos los demás nodos, la suma resultante se invierte y determina la puntuación de proximidad para ese nodo. Es importante destacar que, a pesar de ser unos algoritmos altamente funcionales, estos funcionan de forma óptima en grafos conectados, es decir, que su aplicación en grafos no conectados puede arrojarnos algunos errores de análisis. Pudiésemos obtener una distancia infinita entre dos nodos en componentes conectados por separado. Al ocurrir esto se estaría otorgando una puntuación de centralidad de cercanía infinita cuando hagamos la sumatoria de las distancias de un vértice determinado.

1. **“ALGORTIMOS DE PREDICCION DE ENLACES”**

**ADAMIC ADAR**

Los algoritmos de predicción de enlaces ayudan a determinar la proximidad de un par de nodos. Las puntuaciones calculadas se pueden utilizar para predecir nuevas relaciones entre ellas. La biblioteca de Neo4j GDS incluye los siguientes algoritmos de predicción de enlaces, agrupados por nivel de calidad:

El algoritmo Adamic Adar es una medida utilizada para calcular la proximidad de los nodos en función de sus vecinos compartidos, la función de este algoritmo es predecir enlaces en una red social.

1. **“ALGORTIMOS DE INTEGRACION DE NODOS”**

**FastRP**

Los algoritmos de incrustación de nodos calculan representaciones vectoriales de nodos de baja dimensión en un gráfico. Estos vectores, también llamados incrustaciones, se pueden utilizar para el aprendizaje automático. La biblioteca de ciencia de datos de gráficos de Neo4j contiene los siguientes algoritmos de incrustación de nodos:

En este grupo de algoritmos podemos mencionar el algoritmo FastRP es un algoritmo de incrustación de nodos en la familia de algoritmos de proyección aleatoria. Estos algoritmos están respaldados teóricamente por el lema de Johnsson-Lindenstrauss según el cual se pueden proyectar n vectores de dimensión arbitraria en dimensiones O (log (n)) y aún preservar aproximadamente las distancias por pares entre los puntos. De hecho, una proyección lineal elegida de forma aleatoria satisface esta propiedad.

Por tanto, estas técnicas permiten una reducción agresiva de la dimensionalidad al tiempo que preservan la mayor parte de la información de distancia. El algoritmo FastRP opera en gráficos, en cuyo caso nos preocupamos por preservar la similitud entre los nodos y sus vecinos. Esto significa que a dos nodos que tienen vecindades similares se les deben asignar vectores de incrustación similares. Por el contrario, a dos nodos que no son similares no se les debe asignar vectores de incrustación similares.

El algoritmo FastRP asigna inicialmente vectores aleatorios a todos los nodos utilizando una técnica llamada proyección aleatoria muy dispersa, ver (Achlioptas, 2003) más abajo. A continuación, el algoritmo construye iterativamente incrustaciones intermedias promediando las incrustaciones intermedias vecinas de la iteración anterior o los vectores aleatorios generados durante la primera iteración. En cada iteración, la inclusión intermedia se normaliza utilizando una norma euclidiana estándar. Es decir, cada elemento de la incrustación se divide por la raíz cuadrada de la suma de cuadrados de los elementos emdedding.

Al final, la incrustación resultante para cada nodo es una suma ponderada de las incrustaciones intermedias, donde los pesos son un parámetro de configuración llamado iterationWeights.

Por lo tanto, la incrustación de cada nodo depende de una vecindad de radio igual al número de iteraciones. De esta manera, FastRP aprovecha las relaciones de orden superior en el gráfico sin dejar de ser altamente escalable.

1. **“ALGORTIMOS DE BUSQUEDA DE RUTA”**

**RUTA MAS CORTA DE UNA SOLA FUENTE**

Los algoritmos de búsqueda de rutas encuentran la ruta más corta entre dos o más nodos o evalúan la disponibilidad y calidad de las rutas. La biblioteca de Neo4j GDS incluye los siguientes algoritmos de búsqueda de ruta, agrupados por nivel de calidad:

En este grupo de algoritmos podemos mencionar el algoritmo Ruta mas corta de una sola Fuente este calcula la ruta más corta (ponderada) desde un nodo a todos los demás nodos del gráfico. Este algoritmo es un protocolo de enrutamiento para redes IP. Utiliza el algoritmo de Dijkstra para ayudar a detectar cambios en la topología, como fallas en los enlaces, y crear una nueva estructura de enrutamiento en segundos.

1. **DESARROLLO**
   1. **DESCRIPCION DETALLADA DEL PROBLEMA Y SU PROPUESTA DE SOLUCION.**

Ahora que estamos próximos a las elecciones presidenciales, es difícil saber porque candidato elegir, primero porque hay mas de 10 candidatos, cada candidato con su equipo de trabajo, pero para la ciudadanía es difícil saber por quien elegir, para ello se analizara el nivel de aceptación de cierto candidato en especifico, para ello se ha definido los siguientes problemas y soluciones que se detallan a continuación, para ello nos basamos en la siguiente información.

* Candidato
* Usuarios
* Tweets
* Re-tweets
* Likes
* Localización

Toda la información es obtenida de la red social Twitter del candidato en específico.

**Problema 1**

**Descripción del Problema**

**Solución 1**

1. **Resultados de Análisis**
2. **Opinión y Recomendaciones**
3. **Bibliografía**