Universidad de La Habana
Facultad de Matemáticas y Ciencias de la Computación

Simulación de propagación en Redes Sociales

Integrantes:

Alex Sánchez Saez C412
Carlos Manuel González C411
Jorge Alberto Aspiolea González C412

Abril 2024

1. El problema

La presente investigación consiste en extraer de una red social, simulando el comportamiento básico de las personas en la misma, darle una respuesta a la pregunta que nos planteamos que es: *¿Dado una red de seguidores en una red social, cual es el tipo de post que más alcance tiene en dicha red?* Para resolver esta pregunta vamos a realizar una simulación de dicha red social.

2. Simulación

La simulación consiste en un *medio ambiente*, el cual sería nuestra red social. Este estará representado por un conjunto de posts que abarca diversos temas, formando un vector de características, en el cual, que tan relevante es cada tema en ese post. También los posts serán evaluados por tres estadísticas importantes, *la cantidad de likes*, *la cantidad de dislikes* y *la cantidad de veces que se compartió ese post*.
Por otra parte, las personas serán representadas agentes los cuales podrán tomar decisiones y decidir cómo reaccionar a los post, pudiendo decir entre dar like, dislike o compartir la publicación con otros. Para modelar este sistema de agentes usaremos una **arquitectura multiagentes** en donde los agentes podrán interactuar entre ellos y con el medio ambiente. Estos agentes actuarán basados en la arquitectura **Believes-Desires-Intentions(BDI)**

3. Procesamiento del Lenguage Natural (NLP)

El flujo de simulación empieza por pedirle al usuario algunos datos relacionados al estado de la red social que él desea simular y que objetivos tiene con la simulación. Dichos datos están conformados la cantidad de personas en la red social, una **descripción** de la red social a simular, de donde se extraerán los tos temas más relevantes que se mencionan comúnmente y los objetivos que tiene el usuario con la simulación.
Para esto se usará el procesamiento del lenguage natural, en donde el usuario introducirá un texto mencionando cuantos usuarios tendrá la red, una breve descripción de los temas que se tratan en la misma y una explicación de sus objetivos con la simulación.
Luego de tener el texto usaremos uno **Large Language Model (LLM)** para extraer la información necesaria del input del usuario. Para realizar una extracción más detallada, se realizaron 3 consultas al LLM:

- Una primera consulta en donde simplemente el LLM extrae la cantidad de personas que habrán en la red social.
- Una segunda consulta en donde el LLM analiza el texto y extrae los temas más relevantes que se ven en la red social y que le asigne a cada tema un número indicando el porcentaje de personas que les intereca comúnmente ese tema.
- Y una tercera consulta en donde el LLM analiza el texto y extrae, basado en los objetivos que expresa el usuario, tres números que representan qué tan importante son los **likes**, **dislikes** y **shares** en la simulación para lograr los objetivos del usuario.

Para obtener correctamente los datos del texto del usuario se exigió que el *prompt* tenga al menos 100 palabras para que sea lo suficientemente expresivo, además de usar algunas técnicas de

Prompt Engineering como darle un rol al modelo, especificarle un formato específico para la salida y realizar consultas con tareas simples y directas.

3.1. Importancia de los temas en la red

Una vez se extraen los datos usando el LLM, tocaría llevar los temas proporcionados por el modelo al conjunto de temas que tenemos preestablecidos. Para esto vamos usar la representación en **embeddings** de cada tema (incluido los proporcionados lor el LLM), así podemos hacer una una comparación semántica de cómo de relacionados están dos temas usando la *distancia coseno*. Supongamos que tenemos n temas preestablecidos en nuestra red social, nuestro objetivo es construir un vector v de n elementos donde $v_i \in [0, 1]$ representa que tan relevante es el tema i en la red social. Sea m la cantidad de temas devueltos por el LLM y p_i la importancia dada a cada una en la red. Si tenemos los embeddings de cada grupo de temas(sean a y b respectivamente), entonces la para calcular la relevancia v_i del tema i en la red social basado en los m temas principales devueltos por el LLM podemos calcularlo como:

$$v_i = \frac{\sum_{j=1}^{m} p_j sim(a_i, b_j)}{n}, \forall 1 \leq i \leq n$$

donde $sim(a_i, b_j)$ es la similitud entre los temas i y j usando los embeddings de cada uno.
Esta suma ponderada de la similitud de los temas devuelto por el LLM multiplicados por la importancia que tiene cada una en la red social, nos da una medida aproximada de la relevancia del tema i en la red social.

3.2. Representación del conocimiento

En la simulación, una de las formas de interactuar el agente con el ambiente es mediante la publicación de nuevos posts. Estos posts son generados aleatoriamente basándonos en los **believes**, **desires** e **intentions** del agente. Esto aunque puede funcionar, una forma más realista de hacer que el usuario cree una publicación coherente es mediante una base de conocimiento. Si un agente tubiera acceso a una base de conocimiento en donde, el agente teniendo varios temas de interés puediera seleccionar otros temas acordes a los q él quiere para la publación con un nivel de importancia para cada tema, entonces el usuario será capaz de poder crear publicaciones más abarcadoras en cuantos a temas se refiere, pero sin salirse de los tópicos generales, ya que estaríamos escogiendo otros temas semejantes a los que él ya tiene.
Para esto podemos usar como nuestra "base de conocimiento.ª la matriz de de correlación de los embeddings de los temas. Mas formalmente sería, tener una matriz $M_{n \times n}$ donde $M_{ij} = sim(v_i, v_j) = \alpha, 1 \leq \alpha \leq 1$ indica la semejanza del tema i con el tema j. Con esta matriz de correlación es fácil buscar una un tema, los temas más semejantes a él, y usando el mismo índice de similitud se puede calcular el el nivel de importancia de este nuevo tema en el post.

4. Optimización

Una vez la simulación arroge las estadísticas antes mensionadas, ahora tenemos la interrogante de encontrar *¿qué combinación de temas y en qué cantidad proporciona un mejor alcance en la red?*. Esta interrogante tiene a dos pasos importantes; primero es ¿cómo calificas que una publición proporciona mayor alcance?¿bajo que criterio es interesante realizar una búsqueda de la mejor

solución? y en segundo lugar tenemos el problema de *¿Cómo vamos a encontrar dicha solución óptima?*

4.1. Criterios de optimalidad

Para responder la primera pregunta, podemos ver que cada post en la red social se mide por tres estadísticas principales: *cantidad de likes, cantidad de dislikes y veces compartida*, además que la simulación nos proporciona esas mismas estadísticas enfocándonos esta vez en el tipo de tema, por tanto tendríamos un índice de que tanto gusta cada tema entre los usuarios (cantidad de likes promedio), o cual es el tema que menos gustó (cantidad de dislikes promedio), o incluso cuáles son los temas que más se difundieron, ya sea por gustos o disgustos, en la red(cantidad de veces compartida). Entonces depende de los intereses del tengamos podemos decidir que importancia darle a cada una de esas tres estadísticas.
Para esto lo más interesantes es que, en vez de tener que definir numéricamente que tanto nos interesa una estadística más que otra, definamos en lenguaje natural cuales son nuestras intensiones respecto al tipo de alcance que queremos obtener y de ahí, usando un **LLM** extraer los índices numéricos que más relación tenga con nuestra descripción.

4.2. Encontrar la mejor solución

Para la segunda pregunda de *¿cómo encontrar la mejor solución?*, debemos fijarnos que la solión consiste en un vector x de tamaño n, donde cada componente del vector es un número $x_i \in [0, 1]$ que indica que el tema i debe tener una relevancia x_i en el post para que tenga el mayor alcanze.
Para saber si una solución es mejor que otra debemos tener una forma de cuantificar dicha mejora y así poder hacer la comparación. Para esto definimos una función $f : D \to \mathbb{R}$, donde D es el dominio de todas las soluciones posibles del problema, o sea, todas las formas posibles en las que se puede construir un post, y $f(x) \in \mathbb{R}$ un número que indica que tan buena es la solución. Podemos ver que D es de tamaño infinito, ya que una solución consiste en una distribución de n números entre 0 y 1, y como ese conjunto es no enumerable entonces hay infinitas combinaciones de n numeros que pertenezcan al rango $[0, 1]$.PO
Por ende, una buena forma de enfrentar el problema es usando metaheurísticas para encontrar a la función objetivo f un óptimo que se aproxime a la solución real. Como metaheurística usamos **Particles Swarm Optimization (PSO)**, la que es buena para optimizar funciones.

4.3. Función de optimización