Project – Spreading of information

The main idea of this project is to analyze and assemble end-to-end software engineering project with the goal of describing the news propagation in the Italy with Google Trends. Italy was chosen because it is politically divided in 20 regional regions. Each region reports it's keyword traffic to the Google which is then aggregated and reported as keyword popularity metric.

We are offered sample of the following data, for Italy.

	Title	traffic	link	published
0	manchester city	500+	https://www.dazn.com/en-CA/news/other/pep-guar...	Tue, 20 May 2025 12:20:00 -0700
1	che ci faccio qui	200+	https://www.csvcalabriacentro.it/oggi-su-raitr...	Tue, 20 May 2025 12:20:00 -0700
2	myanmar	100+	https://www.chinadailyasia.com/article/612143	Tue, 20 May 2025 12:20:00 -0700
3	crystal palace vs wolves	200+	https://www.sportinglife.com/football/news/any...	Tue, 20 May 2025 12:10:00 -0700
4	gianluca un posto al sole	200+	https://www.ilmattino.it/schede/un_posto_al_so...	Tue, 20 May 2025 12:10:00 -0700

Each raw in table is called the ***event.***

Events are the football matches because they offer predictable schedule when the game will take place and when key event has happened. We assume that the published article alongside the query is the most popular article which describes or contain observed query.

Analysis further develops itself by calculating the difference between published time and actual event organization.

We are interested in forming the latency table for **event_i:**

Table *communication_latency*
Event *e* (*title*)

Location	Delta (minute)
L1 = *Source*	0
L2	d1
L3	d2
L4	d3

Table *public_latency*
Event e (*title*)

Location	Delta (minute)
L1 = *Source*	0
L2	d1
L3	d2
L4	d3

It is a regionally divided country and each region reports it's own network keyword traffic to the Google.

Model for information spreading: **(event, event_timestamp, published_timestamp)**
For each region r we have a set of keywords: ks = [k1, k2, k3, k4, …] for specific time period t1, t2.

Based on the following we need to create a function:
def bipartite_graph(df: [region, keyword, time]) → output is the [region1, region2, keyword, delta]
 # Function creates combination of all regions which have share same keyword and computes the time difference between them.
Example:
Input:

Region	Keyword	Traffic	Time
R1	Pope	500	12:00
R1	Celebrity walks	100	12:10
R2	Pope	300	12:20
R3	Pope	200	12:30

Output: delta_table

Region1	Region2	Keyword	Delta
R1	R2	Pope	00:20
R1	R3	Pope	00:30
R2	R3	Pope	00:10

delta_table = bipartite_graph(df: [region, keyword, time])
Another function to define is the ***def connect_pairs(delta_table)*** → list which gives us the chained time distance between the source and all the domino pieces along the chain.

delta_table = bipartite_graph(df: [region, keyword, time])
delta_dict = connect_pairs(delta_table)

I1	I2	I3	Time
R1	R2	-	00:20
R1	R3	-	00:30
R1	R2	R3	00:30

Conclusion
This briefing describes information latency extraction of the events which are categorized with the SPORT tag.

Case study:
The *election* and *relection* of pope is instantaneously transferred from one part of world to the opposite part of the world. This means that information and cyberspace is almost immediately filled with the news. What is geopolitical consequence of this phenomena. This means that the news are instantaneously transferred from one place to another through media, videos and other sources. In alternative scenario, the important information would be passed orally or through delegation in order to prevent the informing the foreign agents before the other more relevant source. Which could then act on this information based on his own interest and own logic. This was one way of expression of logic in physical domain which could then be economized.