NAME: LIU,HONGYANG No. 17201091/1

1.The paragraph comes from this article(Shao et al., 2019)

> *Traffic congestions leads to air pollution, green house gas emission and energy consumption in central business district areas. More than one-third of congestions are caused by parking space searching tasks A data-driven and robust solution for parking availability prediction can guide the drivers, thereby reducing traffic congestions and the time cost. However, such solutions cannot be proposed without network infrastructure in the past.*

Answer:

- To extract terms form the paragraphs, we need first build a list of common words as follows: “to, is, and, in, are, by, it, without, the, leads”, etc. The common word list would be considered as “stop words lists” to demonstrate the beginnings or ending of the paragraph.
- Because a term could be defined as a sequence of one or more uncommon words demarcated. We would apply the algorithm as follows:
    - Read the first word of sentence “Traffic”, if it appears in common word list, we would delete it. If is an uncommon word, we would save it.
    - Read the next word “congestions”. If it appears in common word list, we would delete. We would delete it and place the save word(from step one) into the index terms list. If it is an uncommon word, we would append it to the word we save in step one and save them as two-word term(Traffic congestions). If it is a sentence delimiter, place any saved term into the index term list and stop the program.
    - Repeat step one and step.

Follow the algorithms above, in first sentence, we would create index like “Traffic congestions”, “air pollution”

And then, we would parses the whole paragraph sentence by sentence to extract index terms.

References:

Shao, W., Zhang, Y., Guo, B., Qin, K., Chan, J., & Salim, F. D. (2019). Parking Availability Prediction with Long Short Term Memory Model. In S. Li (Ed.), *Green, Pervasive, and Cloud Computing* (Vol. 11204, pp. 124–137). Springer International Publishing. https://doi.org/10.1007/978-3-030-15093-8_9

2.

The first factor: The hdfs replication factor

The default parameter of replication is 3. This means every chunk would be replicated into 3 times. If one of the replica has lost, the nameNode would create another replica to make sure that the number of replicas is equal to 3.

<property>
  <name> dfs.replication <name>
  <value>3<value>
  <description> Block Replication <description>
<property>

We could change 3 to larger number to prevent data loss.

The Second factor: blocks size factor(default 128 MB or 64MB)

We could configure the blocks size value to split the whole file into smaller part.

Each size of block would be equal to the block size value.

<property>
  <name>dfs.block.size<name>
  <value>134217728<value>
  <description>Block size<description>
<property>

The large file would be stored using blocks. These blocks of the file are distributed and would be deployed into the same rack with different positions or different racks. If some machine or racks has been destroyed suddenly, such as power failure and then data loss happened. The other blocks may still work and the NameNode would create the replica of the block.

3.

One approach that ensuring data provenance and trustworthiness is assigning confidence levels to both data and data providers based on the data provenance and trustworthiness.

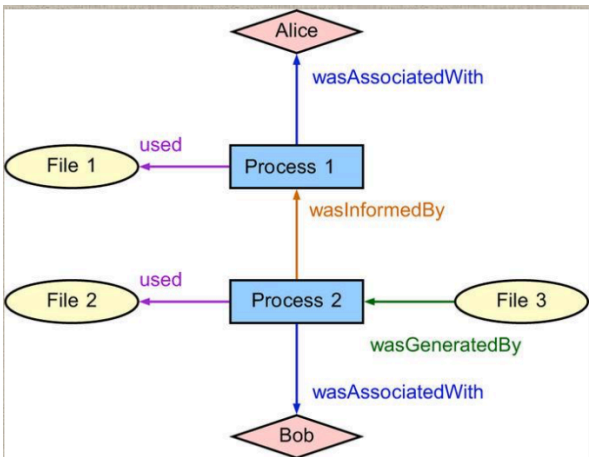
E.g.

We would build a Data Provenance Trust Model to measure three different factors that may affect data trustworthiness, such as data similarity, data conflict, and path similarity.

- In data similarity, we would consider similar items.
- In data conflict, we would consider the same entities with inconsistent descriptions
- In path similarity, we would find out how similar the path of two data items from the sources to destination.

After using this model, we could know if the data sources and the entities handled the data are trust or not.

For example, Process 1 and Process 2 use data from File1 and File2, respectively.



If File 1 has a very high confidence level and File 2 is related with File1. And then File 2 would also have high confidence level because of the path similarity.

Overall, we could assign Confidence Levels to data using Data Provenance Trust Model to ensure data provenance and trustworthiness.