Data Mining & Big Data: A URL Categorization System

Montagner A., Viglianisi E.

University of Trento, 2017

Objectives

Project's objectives:

- Introduce URL Categorization problem.
- Quick discussion on Topic Modeling algorithms.
- Provide a linear (Data Mining approach) and a parallel solution (Big Data approach).
- Show results for both solution.

URL Categorization Problem

STEP 1:

INPUT - A dataset of geotagged URL in M
- A grid of sizes S1xS1 over M

OUTPUT - Main topics for each cell of the grid $S1\text{x}S1$.

STEP 2:

INPUT - A grid of sizes S1xS1 over M, with main topics that were computed in the STEP1.
- A grid of sizes S2xS2 over M

OUTPUT - Main topics for each cell of the grid S2xS2.

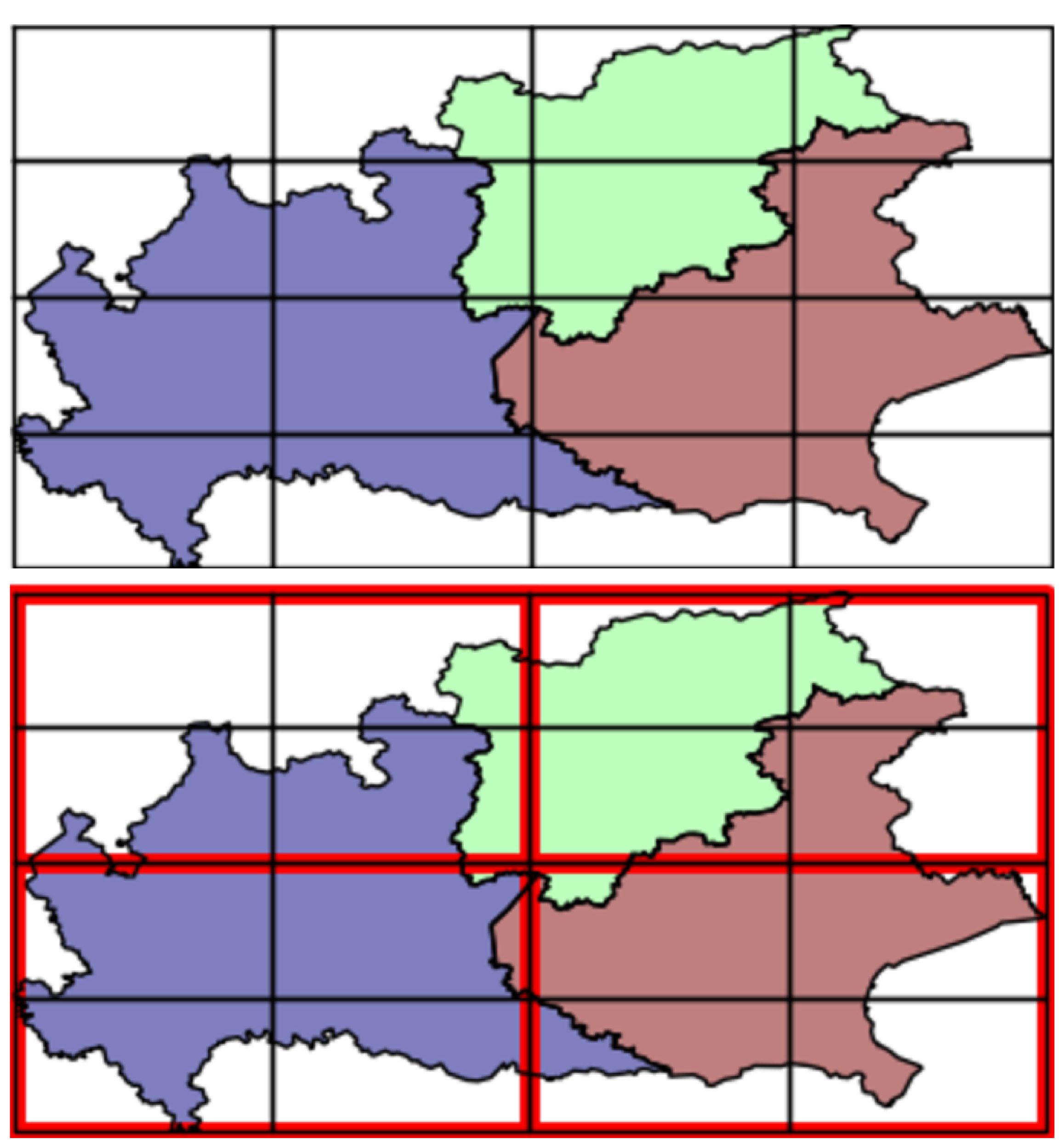

Figure 1: Example of grid division and grid scaling

Topic Modeling algorithms

- lda.update: exploits a method from Gensim library. Its updates an already trained model with new unseen corpus.
- Top_Topic Aggregation: Pre-compute once the LDA model of the entire map (global model), asking for a large number of topics. Then, computes the *top_topic distributions* of the corpus of each subcell. Finally, topic distributions of the subcells are merged to obtain the topic distribution of the entire cell. Allows to save time during the evaluation of top topics: top topics of the new cell are computed by summing together the resulting topics' weights of the already computed sub-cells, instead of iterating again over all the document of the area.

Big Data solution

Since the other two algorithms are very difficult to parallelize, *Top_Topic Aggregation* is the most suitable for this approach because it does not need to maintain an updated shared object.

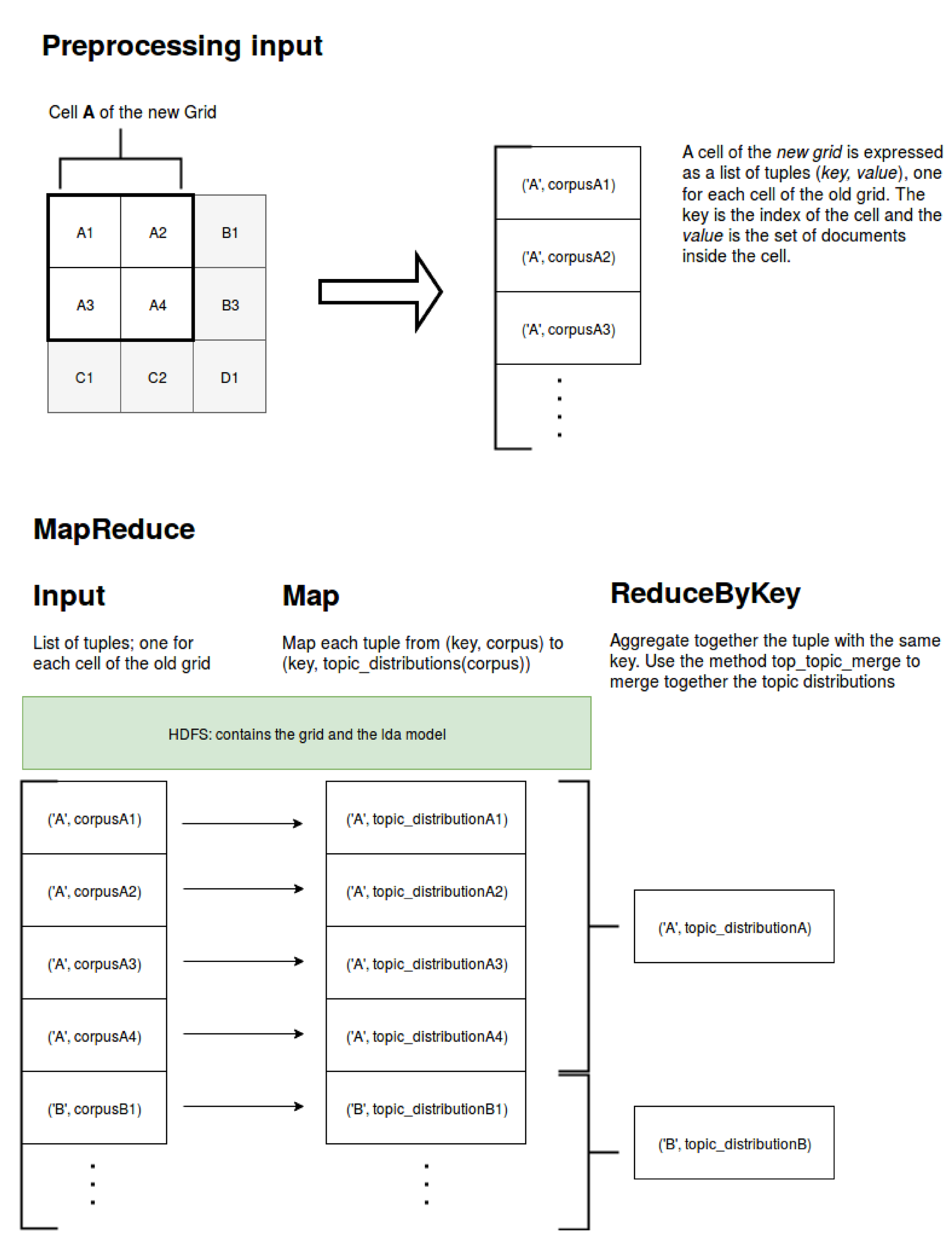

Figure 2: Mapreduce

Data Mining solution

Tests show both strong points and weaknesses of the proposed algorithms.

- *Top_Topic Aggregation*: it is more suitable when the areas are large and every area contains a non negligible portion of the whole dataset.
- *lda_update*: it is the best choice when small increases of the dimensions of the cells are taken in consideration, so that the model will be updated with few documents.

There is no single best way of solving this problem, but rather a combination of the two methods, depending on the required application.

Streaming Spark

MapReduce algorithm is applied for each chunk of data (which spark obtains discretizing the stream) and, when the results for a chunk are ready, they are merged into to the topic distributions of the existing grid.

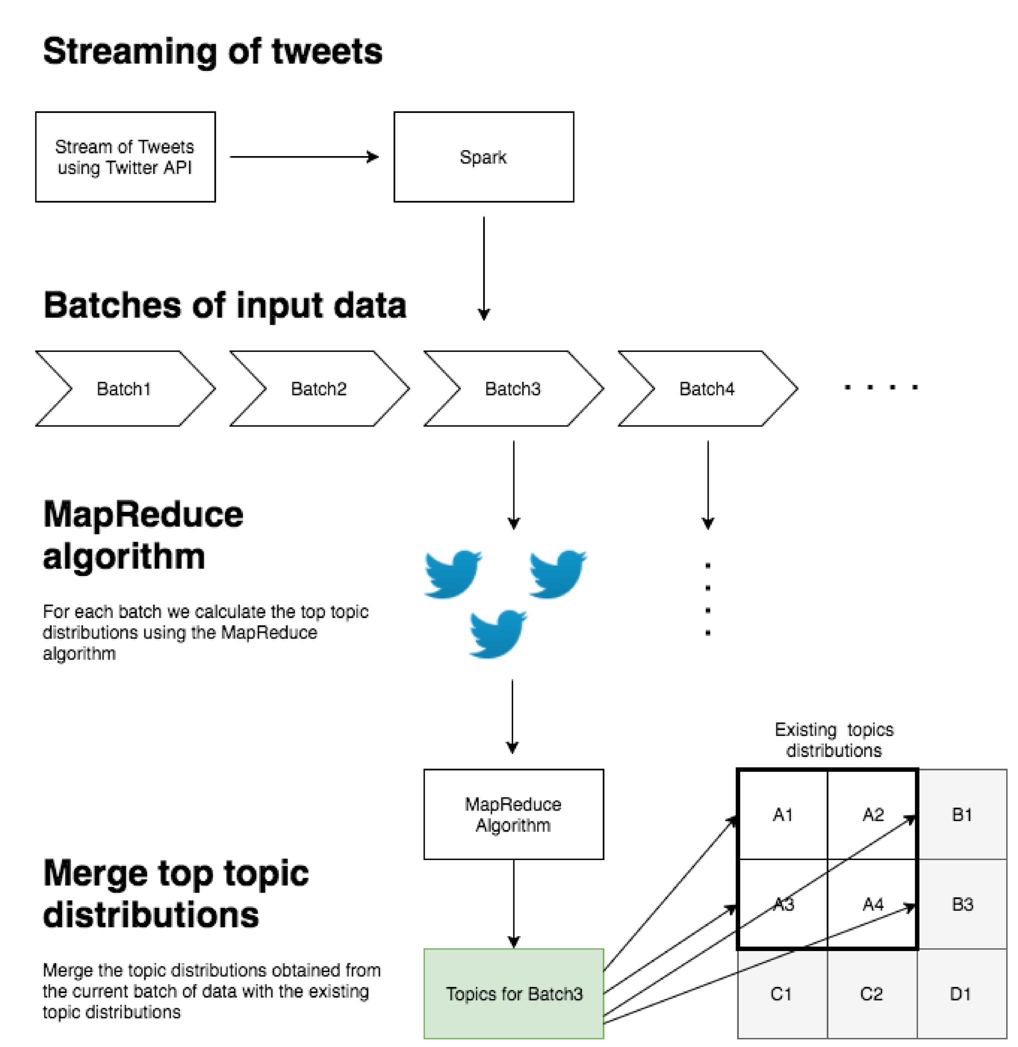

Figure 3: Spark Streaming

Data Mining Results

Table 1: My caption

Documents	*Top Topic Aggr*	*lda.update*
5056	*6*	*7*
14329	*10*	*10*
430	*1*	*3*

Big Data Results

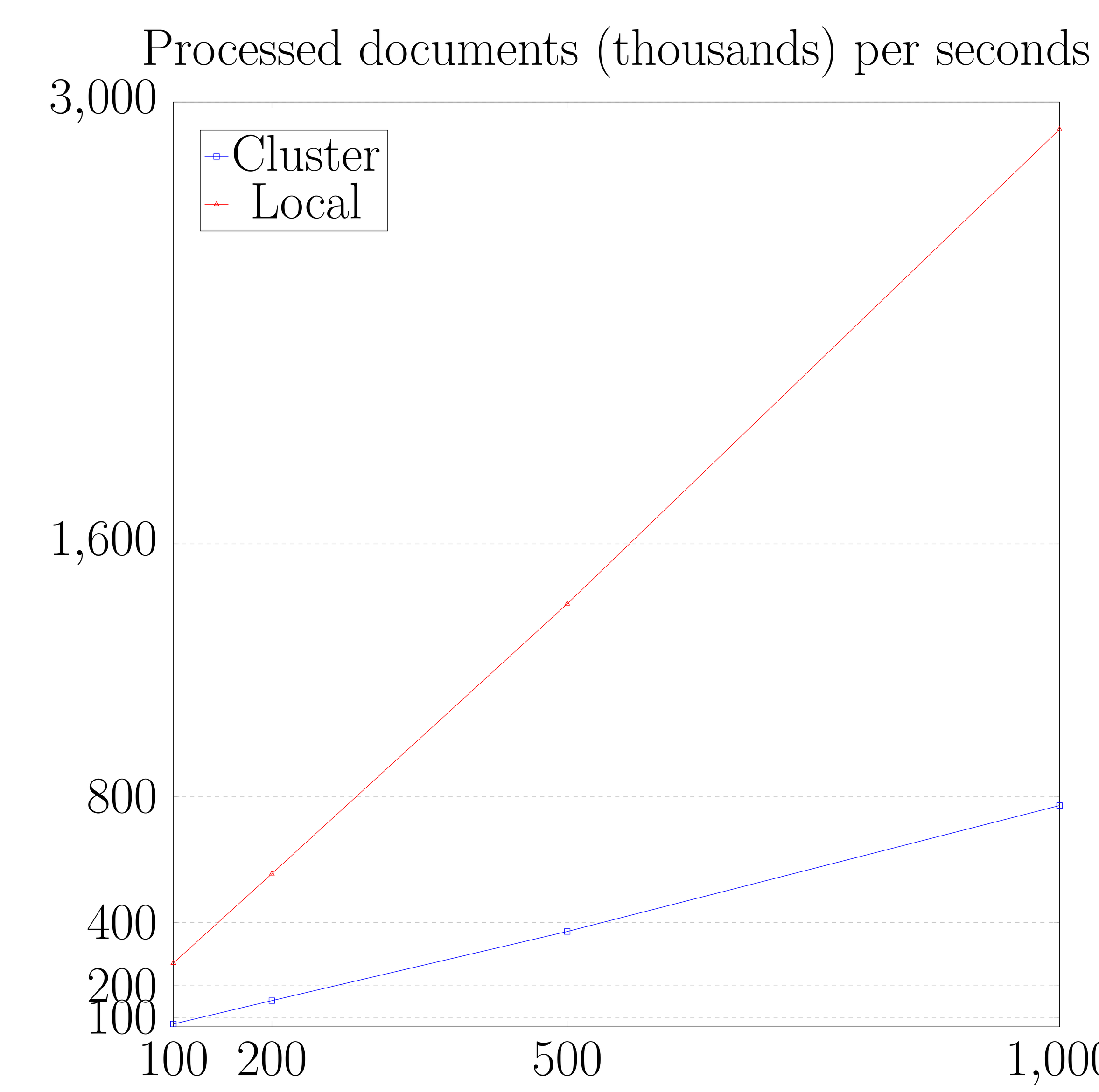

Table 2: **Cluster Results (4 machines, 4 cores)**

Number of documents	*Computation time*
20k	*20s*
100k	*70s*
200k	*153s*
500k	*372s*
1000k	*771s*

Table 3: **Local Results, (1 machine, 4 cores)**

Number of documents	*Computation time*
20k	*50s*
100k	*272s*
200k	*555s*
500k	*1410s*
1000k	*2913s*