LARGE SCALE DISTRIBUTED SYSTEM FOR SENTIMENTAL ANALYSIS

Lexicon Builder:

The lexicon builder constructs Twitter polarity lexicon from a list of positive and negative seed words that are emotional icons (emoticons) such as ":)", ":D", ":(", ":-(", etc. In the original algorithm, the authors first compute the co-occurrences for each pair of words/phrases from a collection of Web documents. A cosine similarity score between two arbitrary words/phrases is derived from their co-occurrences. A word graph is then constructed with each node representing a single word or a phrase. Two nodes are connected by an edge with a weight that equals the cosine similarity score between two words/phrases represented by these nodes. Edges having weights below a certain threshold are discarded. Sentiment scores are propagated from seed words, which are initially positive and negative words, into other nodes. This preprocessing step is done so as to reduce the total nodes in the graph.

Generating co-occurrence matrix-A MapReduce job is used for computing co-occurrences between words/phrases in the training tweets. Phrases are formed by connecting continuous words. In our experiment, we only used bi-gram phrases.

- Computing cosine similarity and creating the graph of words-The cosine similarity between two words Wi and Wj is computed by dividing the inner product of two vectors wi and wj with coordinates (ki1, ki2, …, kin), (kj1, kj2, …, kjn) respectively, by the product of their lengths. In order to avoid re-computing vector lengths, all vectors wi are first normalized to unit vectors w'i , then the cosine similarity values are calculated as the pairwise dot product between the unit vectors w'i

- Removing edges with low weights -The purpose of this step is to keep only edges with top weights so that we can avoid non-sentiment words and improve the speed of sentiment score propagation. An edge (wi , wj) is kept if it is one of the TOP_N highest weighted edges adjacent to nodes wi and wj .

- Sentiment Score propagation - This is the main algorithm which propagates sentiment scores from seed nodes into other nodes which are located within a distance D from them. Score propagation is implemented with the help of an HBase table named "propagate". In this table, nodes reachable from a seed node are stored as qualifiers in a column family named "visited" in the row with the key equal to the seed node's id.

- Sentiment Score classification - After the sentiment score propagation step is completed, each node wi in the graph will be assigned two types of sentiment scores: score - sum of scores propagated from positive seed nodes, score - sum of scores propagated from negative seed nodes.

  This step is executed with three MapReduce jobs. The first job is launched for calculating score and score of each node. The second one calculates β. The last one computes score