Assignment 2

Rajeev Kumar
Department of Computer Science
Indian Institute of Technology Kanpur
rajeevks21@iitk.ac.in

Divyansh Chhabria
Department of Computer Science
Indian Institute of Technology Kanpur
divyanshc21@iitk.ac.in

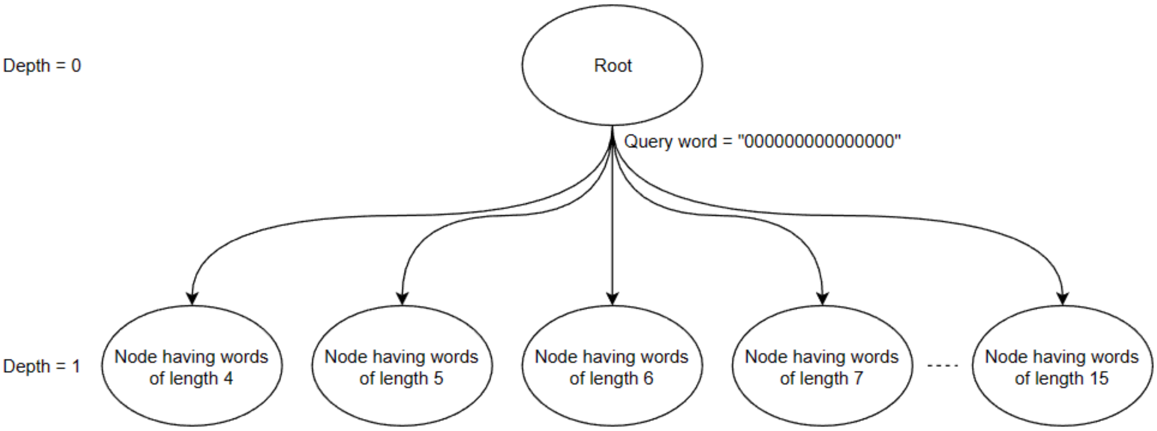
Sandeep Nitharwal
Department of Computer Science
Indian Institute of Technology Kanpur
nsandeep21@iitk.ac.in

Praneat Data
Department of Mathematics
Indian Institute of Technology Kanpur
praneat21@iitk.ac.in

Vrinda Sharma
Department of Mathematics
Indian Institute of Technology Kanpur
vrindas21@iitk.ac.in

1 Node splitting criterion

At root, we passed ”000000000000000” (15 times 0’s) as the query word. As this string does not belong to dictionary, using this as the query word divided all the words into nodes depending upon the length of the word (number of letters in the word). Hence, in each bag formed at depth 1, words having the same length are in the same bag.


Depth = 0
Root
Query word = "000000000000000"
Depth = 1
Node having words of length 4
Node having words of length 5
Node having words of length 6
Node having words of length 7

Node having words of length 15

At every other node, we implemented a greedy algorithm to split the node. The algorithm used aims to minimize entropy while being efficient.

We know that the entropy of a collection of bag of words $\{B_1, B_2, B_3, ..., B_k\}$, $H(B_1, B_2, B_3, ..., B_k) = \sum_{i \in [k]} \frac{n_i}{n} ln(n_i)$ where n_i is the number of words in the bag B_i.

We know that the more the number of nodes is formed, the more information is gained. So, we maximized the number of nodes formed.

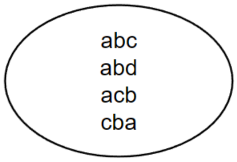
An approximate algorithm to find a query word that leads to the formation of the maximum number of nodes is to find the word whose letters match the most with the corresponding letters of other words in the node.

Using such a query word leads to the formation of the maximum number of nodes because of its similarity with other words in the node. Note that it may be the case that this word, when used as a query, may not result in the formation of maximum nodes, and some other word may do that. But using such a word as the query promises us that the tree formed would not be imbalanced greatly, and the tree distribution lies closely to the distribution we would have obtained if we would have used the word that gives maximum nodes upon being queried.

Algorithm used to find the query word:

- For each node (except root), form a $2D$ matrix M of size $26 \times (size\ of\ words\ in\ the\ class)$.
- Initialize the matrix with zeros.
- Each entry M_{ij} of this matrix, at the i^{th} row and the j^{th} column represents the number of occurrences of the letter numbered i at position j in the word.
- Iterate over each word in the node and over each letter of the chosen word and update the matrix.
- After updating the matrix for every word in the node, iterate over each word again, and using the matrix formed, we find the score of each word in terms of how much its letters match with corresponding letters of other words in the node. This is done by simply adding the count of matches of each letter in the word using the matrix formed.
- The word with the maximum score is selected as the query word. In case more than one has the maximum score, the word first appearing in the node is selected as the query word.

Example:
Suppose there is a node having the following words:- {abc, acb, cba, abd}. Then the matrix for this is:-



	0	1	2
0	3	0	1
1	0	3	1
2	1	1	1
3	0	0	1

All the other entries of the matrix are 0 because no other letter is present in the given words.
0^{th} row corresponds to 'a', 1^{st} row to 'b', and so on. The value in the 0^{th} row and 0^{th} column is equal to 3 because 'a' occurs three times at 0^{th} position ('a'bc, 'a'bd, 'a'cb). Similarly, the value in the 0^{th} row and 1^{st} column is equal to 0 because 'a' does not occur at 1^{st} position in the given words, and so on. The score of the word "abc" is equal to $3 + 3 + 1 = 7$, "abd" is equal to $3 + 3 + 1 = 7$, "acb" is equal to $3 + 1 + 1 = 5$, and "cba" is equal to $1 + 3 + 1 = 5$. The maximum score is 7, corresponding to the words "abc" and "abd". Since, "abc" occurs before "abd", "abc" is chosen as the query word.

2 Criterion to make a node leaf node

We made a node leaf node if the $size\ of\ the\ node \leq 1$ or the $depth \geq 15$.

3 Hyperparameters and pruning strategies

We did not make use of any hyperparameter or pruning strategy.

4 Results obtained on training data

- *Average number of queries made* : 4.059
- *Training time* : 0.2061 *seconds*
- *Model Size* : 1089643
- *Accuracy* : 100%