The TRSL Project

Internship at IISc, Bangalore. Implement a research paper, involving decision trees for language modeling.
Interns: Kshitij Karthick, Ravi Kumar L

Sponsor: Prof. T V Sreenivas, Speech and Audio Group, IISc
Mentor: Kiran Subbaraman

Project: https://github.com/iisc-sa-open/trsl

Agenda

- Purpose of this work
- Core algorithm – Decision tree building, and predicting the target
  - Tree construction
    - What the algorithm is; How it is build & Why it was built this way
  - Set construction
- Observations

Purpose

- Algorithm Implementation
  - This algorithm is based on the paper: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=32278
    - “A Tree-Based Statistical Language Model for Natural Language Speech Recognition”, (1989) by L R Bahl, P F Brown, P V de Souza, R L Mercer
  - Use Python, and its related libraries to build a usable implementation of the algorithm

“What” is the Algorithm

- **Step 1:** For every leaf node in the tree, initially the root
  - **1.a:** Ask questions, which have not been asked before (in this lineage)
  - **1.b:** Identify the question which results in minimum *average conditional entropy*
- **Step 2:** Select the leaf-nodes, from the tree, which do not result in an '*over-fit*'
  - **2.a:** From these leaf nodes, choose the node, which minimizes the average conditional entropy.
    - **2.a.i:** Is the stopping criteria met
      - **2.a.i.a:** If not, For this leaf-node create two child nodes that represent the 'Y', and 'N' paths. Go to Step 1, for the child nodes
      - **2.a.i.b:** If yes, tree construction is complete
- **Step 3:** Get a probability distribution for the target at every leaf node

Step 1.a: Ask questions not asked before, in this lineage

1. Training Samples, as a n-gram table

X1	X2	Target
how	was	the
how	was	it
why	was	the
where	is	a
where	are	the
where	was	I

2. Vocabulary grouped into disjoint Sets

s1	{how}
s2	{where, why}
s3	{was, are, is}
s4	{the, a}
s5	{it, I}

3. Words replaced with their set identity

X1	X2	Target
s1	s3	s4
s1	s3	s5
s2	s3	s4
s2	s3	s4
s2	s3	s4
s2	s3	s5

4. Questions of this form are asked:

X1 = s1 ?
X2 = s2 ?
X1 = s2 ?

Step 1.b: Min. Avg. Conditional Entropy

$$\bar{H}_c(Y|X_i = k) =$$

$$- Pr\{X_i = k \mid c\} \sum_{j=1}^{n} Pr\{target = j \mid c\}$$

$$. \log_2 Pr\{target = j \mid c, X_i = k\}$$

$$- Pr\{X_i \neq k \mid c\} \sum_{j=1}^{n} Pr\{target \neq j \mid c\}$$

$$. \log_2 Pr\{target \neq j \mid c, X_i \neq k\}$$

- The average conditional entropy at a node c due to the question Xi = k ?
- **n is the number of sets**
- Pr {target = j | c} is the probability that the target is j given that the immediate set history leads to c in the decision tree

Step 2: Avoid Overfitting

- Overfitting happens in a decision tree when the tree perfectly mirrors the training samples.
- This can be avoided by not splitting a node when too few training samples reach one or both of the tentative child nodes.

Step 2.a.i: Stopping Criteria

$$\frac{Average\ entropy\ of\ leaf\ distributions}{Entropy\ at\ root} > S$$

- S is a tunable parameter.
- Represents the fraction of entropy we want to reduce on an average before a prediction in our model.

$$\bar{H}(Y) = \sum_{i=1}^{L} H_i(Y) \cdot Pr\{l_i\}$$

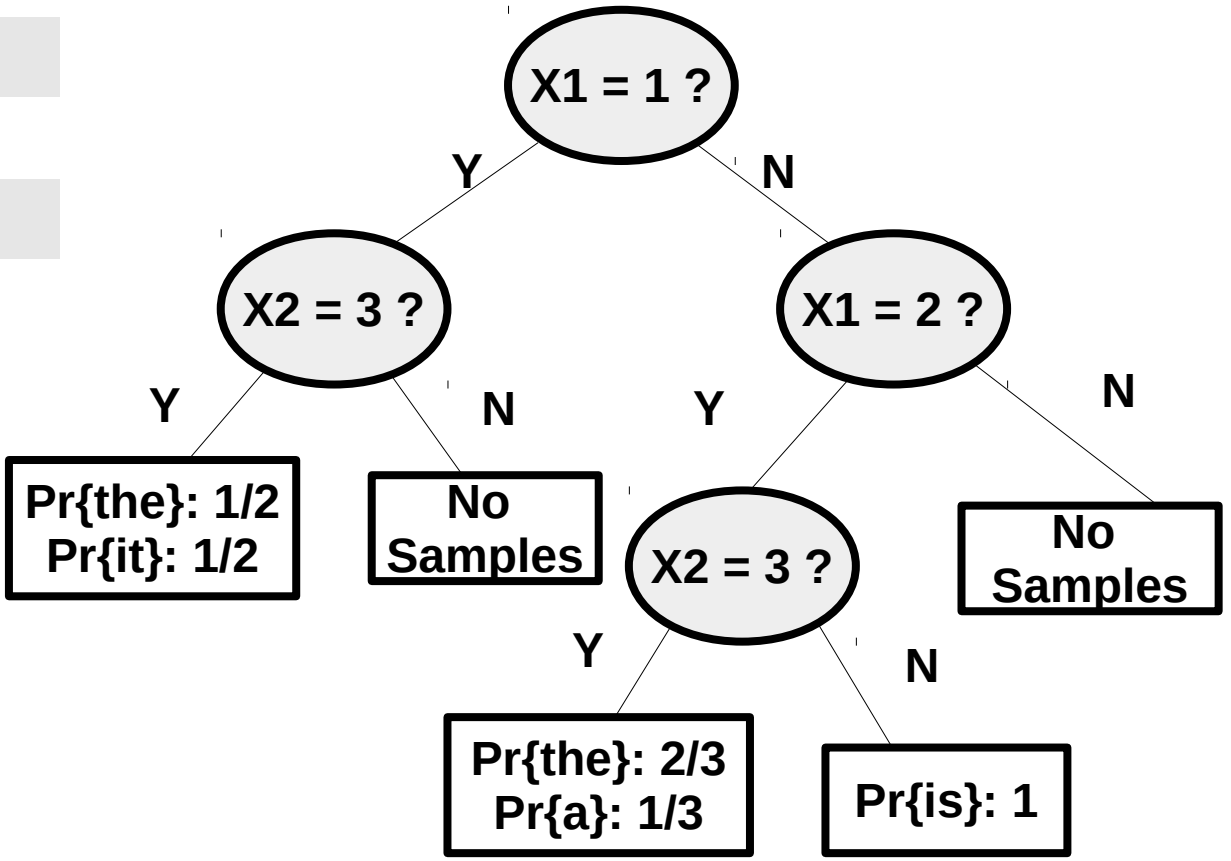
- Average entropy of leaf distributions
- L is the number of leaf nodes
- Hi(Y) is the entropy at leaf node li
- Pr{li} is the probability of reaching leaf node li

$$H_i(Y) = -\sum_{j=1}^{n} Pr\{target = j \mid l_i\} \cdot \log_2 Pr\{target = j \mid l_i\}$$

- n is the number of sets
- Pr{target = j | li} is the probability that the target is j given that the immediate history of sets leads to the leaf node li

Step 3: Probability dist. for target at every leaf node

X1	X2	Target
how, 1	was, 3	the, 4
how, 1	was, 3	it, 5
why, 2	was, 3	the, 4
where, 2	is, 3	a, 4
where, 2	are, 3	the, 4
where, 2	it, 5	is, 3



- ***Traverse*** the tree built with each of the training samples.
- At each leaf node, measure the probability distribution of the target of all the samples that reach that node.

Set Construction

- Intent: Grouping the vocabulary into disjoint sets of words
  - The sets, in combination with the predictor variables, help define questions to ask at every node
  - The sets are related words grouped together.
    - Word2Vec – discovers relatedness in a corpus. Word2Vec provides a representation of words in vector form.
    - K-means clustering – helps us create groups of these related words. These clusters are the sets we use for generating questions.

Observations

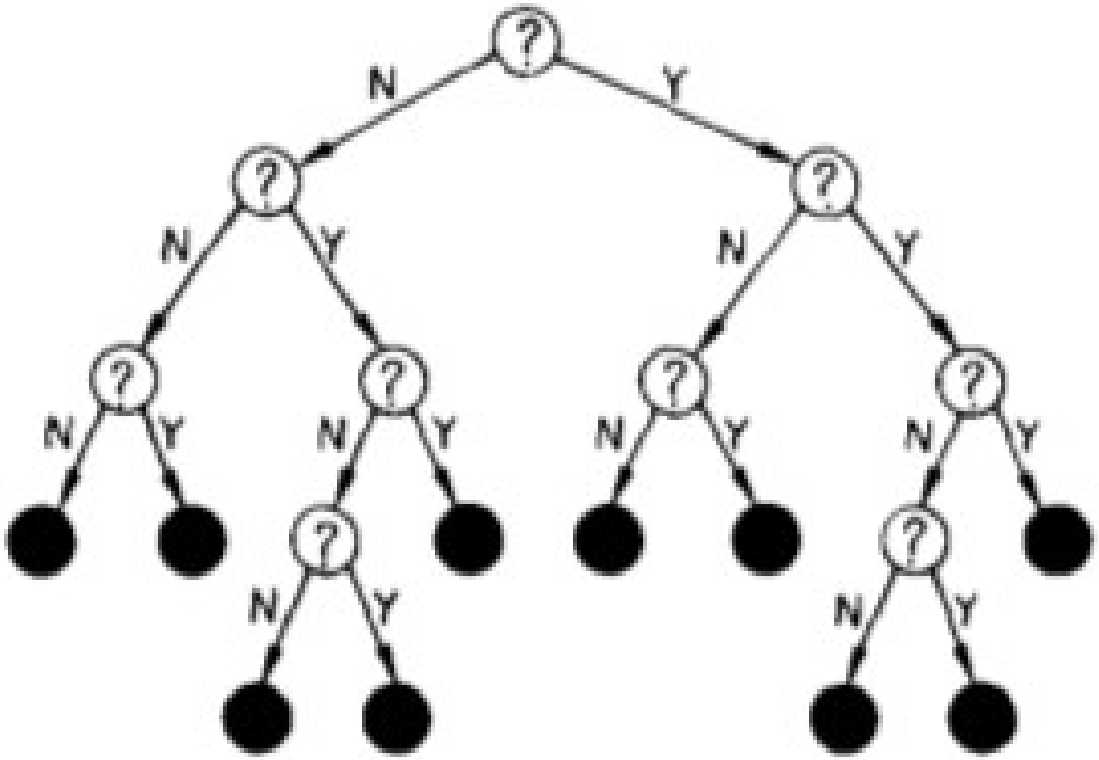
- What we measured
  - Input parameters – number sentences in the corpus, ngram, number of sets, size of vocabulary
  - Measurements – number of leaf nodes in the tree, entropy at root node, Depth at which there is a 30%, and 60% reduction in entropy from the root node
- Explore Data: To arrive at heuristics, or hypotheses
  - Link: https://cdn.rawgit.com/iisc-sa-open/trsl/master/visualization/index.html

Backup slides

Terminology

- Memory (Not Computer memory)
- Sliding window
- Word Sequence
- Sets
- Set Sequence
- Predictor Variables (X1, X2, ... Xm)
- Target

Decision Tree



Lineage?

Binary Decision Tree. Source: Tree based statistical language model paper