Brown Clustering

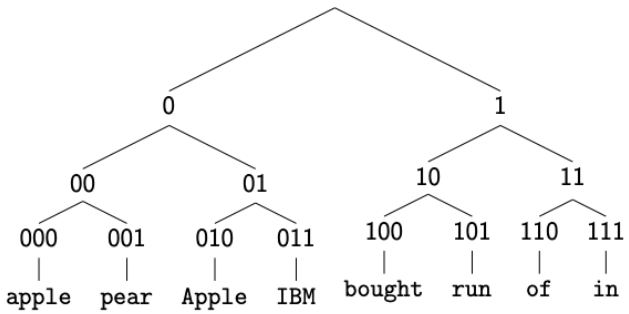
Anumeha Agrawal
anumeha2
University of Illinois at Urbana-Champaign

Introduction

At a higher level, Brown clustering algorithm is an hierarchical clustering algorithm where given a large corpus of words we partition words that have similar context into the same cluster. Now the question arises, what exactly is hierarchical clustering? Hierarchical clustering groups similar objects together by first considering an observation as a separate cluster and then repeatedly identifying its two closest clusters and merging them. It then forms a dendrogram which captures the hierarchical relationships between clusters. Brown algorithm is based on n-gram mutual information and is calculated using distribution information of groups. For a fixed task and corpus genre, the hypothesis is that each corpus size has an optimal c and each c has an optimal corpus size. We eventually notice that the algorithm scales well eventually with new word types.

Algorithm

Input: a sequence of words
Output: for each word type, its cluster which is represented as a cluster represented as a bit string. The image below represents how the output of Brown clustering looks like in terms of bit string.



The formulation

V is the set of all words seen in the corpus w1, w2,...wn
C : V → {1, 2,...k} is a partition of the vocabulary into k classes. Each word in the vocabulary is assigned to a cluster that is represented by an integer.

The model:

p(w1, w2,...wn) = Πe(wi|C(wi))q(C(wi)|C(wi-1))

Here the function C is deterministic and each word is mapped to a state. E and q are two parameters in this brown algorithm. Q represents the probability that given a cluster C(wi-1) the probability of choosing the next cluster.

C(w0) is a special start state

Running the algorithm

We begin with |V| clusters where each word gets its own cluster

At the end of the algorithm we need to get final **k** clusters

We run |V| - k merge steps

- At each merge step we pick two clusters ci and cj , and merge them into a single cluster.
- We use a greedy algorithm to pick a cluster at each step based on the quality of a cluster.

Applications of Brown Clustering algorithm

1. One of the applications of Brown clustering is a Twitter part-of-speech tagger which makes use of several new features and large scale word clustering. We use brown clustering to create clusters that are hierarchical in a binary tree. Here each word is associated with a bitstring (tree path) with length ≤16, and prefixes of the bitstring are used as features. The features that were clubbed by these clusters were successful in identifying different word types and emotions and assigning them prefixes which resulted in apt features.
2. Brown clustering can be used as part of a language model, in tasks such as spell checking and speech recognition, or text features in various NLP tasks such as named entity recognition or machine translation.
3. Brown clustering can also be used to enhance parsing performance for morphologically rich languages which have unlexicalized text.
4. There are different softwares that can be used to implement brown clustering in various applications :-
    a. https://github.com/percyliang/brown-cluster
    b. http://research.microsoft.com/en-us/downloads/0183a49d-c86c4d80-aa0d-53c97ba7350a/default.aspx

Conclusion

Brown clustering is an unsupervised method for learning distributional representations of words through hierarchical clustering and is emerging as an important technique for various NLP tasks like named entity recognition and PoS tagging. It is also being used as corpus word representations through different encoding. One of the main differences between K-means algorithm and hierarchical clustering is that k-means clustering requires some prior knowledge of K which is the final number of clusters the data would be divided into. However, in hierarchical clustering we make the decision to stop merging and determining the final number of clusters by interpreting the dendrogram. Hierarchical clustering can also not handle data as large as k-means can because of its time complexity which is O(n^2).

References

https://aclanthology.org/R15-1016.pdf
https://aclanthology.org/P10-1040.pdf
https://aclanthology.org/J92-4003.pdf
http://aritter.github.io/courses/5525_slides/brown.pdf
https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs190870
http://www.cs.cmu.edu/~ark/TweetNLP/owoputi+etal.tr12.pdf
https://arxiv.org/pdf/1608.01238.pdf
https://www.linguistics.rub.de/konvens16/pub/9_konvensproc.pdf