Technology Review
Sicheng Jiang
sj62@illinois.edu

Brown Clustering

I. Introduction

Brown Clustering is a text clustering algorithm used for assigning each word of a text into a cluster. It forms clusters based on the meaning and syntactic function of the words. It uses a bottom-up progressive (agglomerative) approach so that the output is a hierarchy of clusters where the similar words are combined together to form a cluster closer to the bottom of the tree. [1]

II. Brown Clustering Model

The intuition of this algorithm is that the word which has similar context are likely to have similar meaning or syntactic function. For example, when we observed the sentences “Today is Monday.” and “Today is Tuesday.”, we can concluded that “Monday” and “Tuesday” is likely to have similar syntactic function, therefore we can assign them into the same cluster.

More formally, given a training text t of length T, we want to find a function π mapping a vocabulary of V words into C classes. And we assume the following language model: for each word w_i, $Pr(w_i|w_1w_2\dots w_{i-1}) = Pr(w_i|c_i)Pr(c_i|c_{i-1})$, where $c_i = \pi(w_i)$ and $c_{i-1} = \pi(w_{i-1})$.

We define the quality of a partition π as

$$Q(\pi) = \sum_{w_1w_2} \frac{1}{T-1} logPr(w_1w_2) = \sum_{w_1w_2} \frac{C(w_1w_2)}{T-1} logPr(c_2|c_1)Pr(w_2|c_2) \tag{1}$$

Notice that $Q(\pi)$ is proportional to the log likelihood of generating the training text under 2-gram language model assumption. Simplifying (1), we get

$$Q(\pi) = \sum_{c_1c_2} \frac{C(c_1c_2)}{T-1} \log \frac{Pr(c_2|c_1)}{Pr(c2)} - H(w) \tag{2}$$

where $H(w)$ is the entropy of the word distribution, and therefore do not change with π. The first term in equation (2) is the average mutual information of adjacent classes. [1] Therefore, by maximizing $Q(\pi)$, we are actually maximizing the average mutual information of each pair of adjacent classes, which makes sense intuitively: given the class of the previous word, we want to know the class of the current word as confident as possible.

III. The Clustering Algorithm

There are two approaches mentioned in [2]. But notice that both are approximate suboptimal approach, since the optimal algorithm maximizing mutual information is unknown [1].

a. The First Approach

Assign each word in the vocabulary to its own cluster. Then find a pair of cluster (c_i, c_j), such that merging them will result the largest $Q(\pi)$. Repeating this process until $Q(\pi)$ does not increase. The optimized run time is $O(|V|^3)$, therefore it can only be used for small inputs.

b. The Second Approach

Assign m most popular words to its own cluster (where m is a parameter). Assign the next popular words to its own cluster, then merge a pair of cluster such at $Q(\pi)$ is maximized. Repeat until $Q(\pi)$ does not increase. The optimized run time for this approach is $O(|V|m^2 + n)$ where n is corpus length [2].

IV. Comparing to other text clustering algorithms

a. Comparing to EM

Both Brown and EM are probabilistic clustering models. The difference is that Brown clusters words while EM clusters documents. Brown uses a agglomerative approach while EM starts with an initial tentative clustering and iteratively improves it [3]. Also, EM assigns documents to clusters softly with a probability to multiple clusters, while Brown uses a definite function map each word to exactly one cluster.

B. Comparing to HMM

If we treat latent variables in HMM as clusters and observed variables as words, then HMM can also be used to perform text clustering, and looks very similar to Brown Clustering because both assumes $Pr(w_i|w_1w_2\dots w_{i-1}) = Pr(w_i|c_i)Pr(c_i|c_{i-1})$. The difference is that HMM allows multiple clusters to generate the same word with positive probability, while Brown only allow a word to be generated by exactly one cluster. Also, the object function of the two approaches are different. HMM maximizes the probability of the observed sequence [4], while Brown maximizes the mutual information of adjacent classes.

V. Tools

Here are some open source tools that implement Brown Clustering:

1. brown-clustering • PyPI
2. percyliang/brown-cluster: C++ implementation of the Brown word clustering algorithm. (github.com)

VI. Conclusion

Brown clustering algorithm takes a text string as input and outputs a hierarchy of clusters using a probabilistic agglomerative approach. We discussed the language model it uses and two greedy algorithms to perform the clustering. We also discussed the similarities and differences between Brown Clustering and some well-known algorithm like EM and HMM.

VII. References

[1] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, Jenifer C. Lai. “Class-Based n-gram Models of Natural Language” https://aclanthology.org/J92-4003.pdf
[2] Michael Collins. “Brown Clusters” http://aritter.github.io/courses/5525_slides/brown.pdf
[3] CS410 Text Information Systems Lecture Week 10 https://www.coursera.org/learn/cs-410/lecture/PsyKR/10-5-text-clustering-similarity-based-approaches
[4] Hidden Markov model https://en.wikipedia.org/wiki/Hidden_Markov_model