Word Embeddings

Brown Clustering

Brown Clustering

- Input:
  - A lot of (unlabelled) text
- Output:
  1. An assignment between a word and exactly one cluster
  2. A hierarchy between these clusters

Brown Clustering –Example by Brown 1992

Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays
June March July April January December October November September August
people guys folks fellows CEOs chaps doubters commies unfortunates blokes
down backwards ashore sideways southward northward overboard aloft downwards adrift
water gas coal liquid acid sand carbon steam shale iron
great big vast sudden mere sheer gigantic lifelong scant colossal
man woman boy girl lawyer doctor guy farmer teacher citizen
American Indian European Japanese German African Catholic Israeli Italian Arab
pressure temperature permeability density porosity stress velocity viscosity gravity tension
mother wife father son husband brother daughter sister boss uncle
machine device controller processor CPU printer spindle subsystem compiler plotter
John George James Bob Robert Paul William Jim David Mike
anyone someone anybody somebody
feet miles pounds degrees inches barrels tons acres meters bytes
director chief professor commissioner commander treasurer founder superintendent dean custodian

https://www.aclweb.org/anthology/J92-4003.pdf

Brown Clustering – The model

➔We are looking for the sequence of clusters $c_1 \dots c_n$ given a sequence of words $w_1 \dots w_n$

$$\max_{\text{cluster\_sequences}} p(c_{1:n} | w_{1:n})$$

➔Most N-grams will have counts of 0 …

Browns approach:

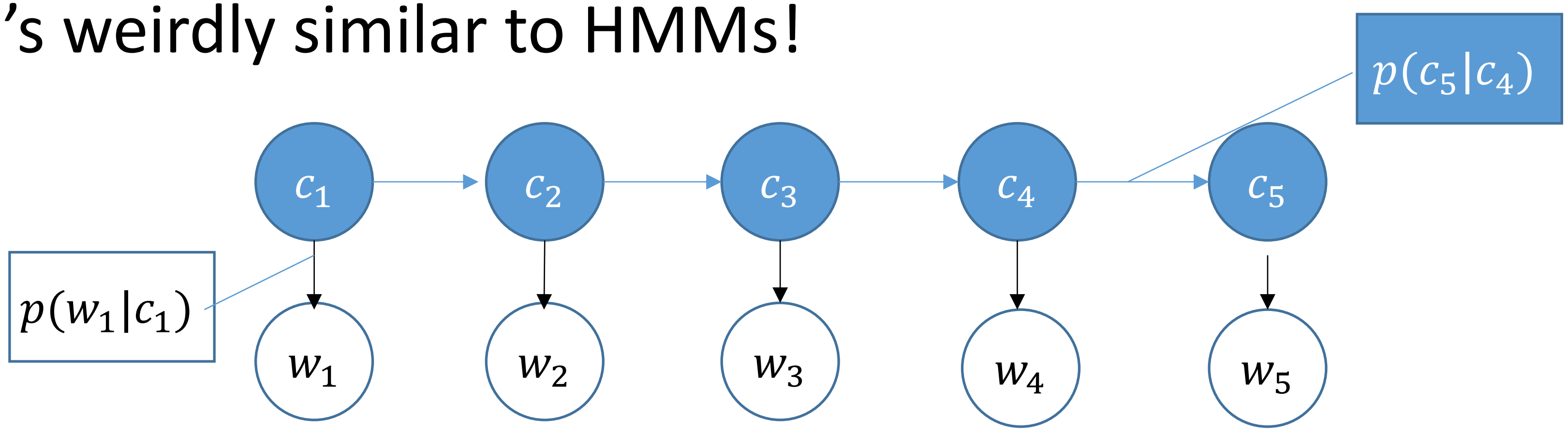
1. Each word w_i originates from only a single cluster c_i, and is only dependant on c_i
2. The cluster c_i does only depend on the cluster c_{i-1}, the cluster of the previous word

Brown Clustering – The model

- Let us apply his approach to the probability:

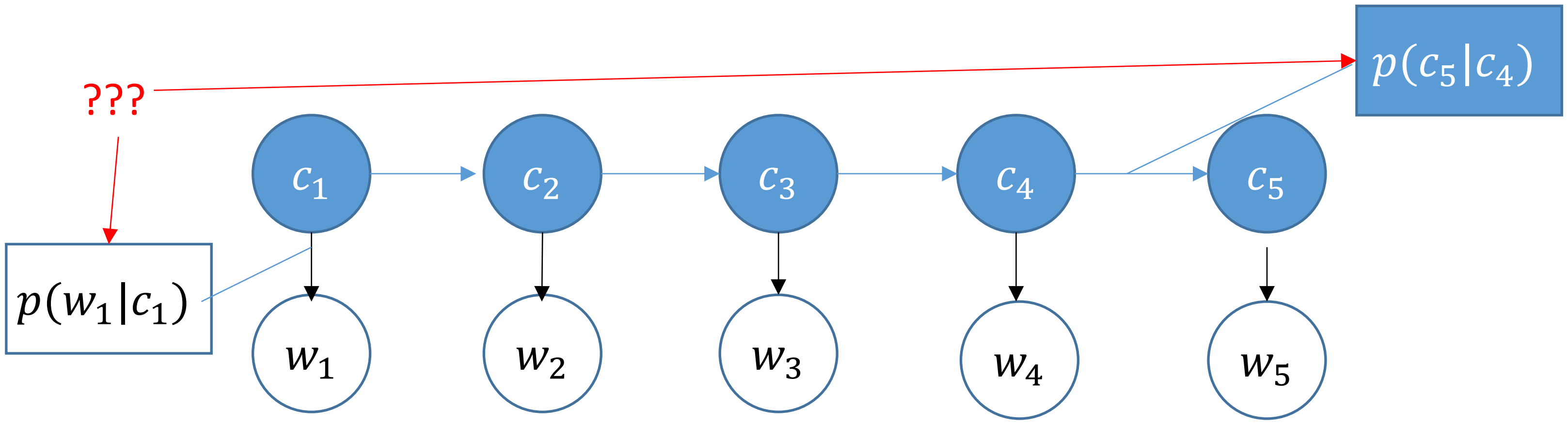
$$p(c_{1:n}|w_{1:n}) \approx \prod_{i=1}^{n} p(c_i|c_{i-1}) \cdot p(w_i|c_i)$$

- That’s weirdly similar to HMMs!



Brown Clustering – The model

- But it's not quite classical HMMs:
  - Each word originates from exactly one cluster
  - We do not have a corpus at hand to read the required probabilities!



Brown Clustering-Algorithm

- Brown applied an iterative approach:

  1. Start by assigning a distinct cluster to every word (type)
  2. Merge the two clusters, which maximize a **given objective function** across an entire corpus
  3. Repeat step 2, until only a single cluster remains

→ This produces a binary tree, where words are leaves

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \frac{1}{m} \log \prod_{i=1}^{n} p(c_i|c_{i-1}) \cdot p(w_i|c_i)$$

➔ That’s the HMM alike formulation we have seen already!

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

Can now be counted!

$$Quality(C) = \frac{1}{m} \log \prod_{i=1}^{n} p(c_i|c_{i-1}) \cdot p(w_i|c_i)$$

We will need that later!

Numerical stability

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \frac{1}{m} \log \prod_{i=1}^{n} p(c_i|c_{i-1}) \cdot p(w_i|c_i)$$

$$= \frac{1}{m} \sum_{i=1}^{n} \log p(c_i|c_{i-1}) \cdot p(w_i|c_i)$$

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \frac{1}{m}\sum_{i=1}^{n} \log p(c_i|c_{i-1}) \cdot p(w_i|c_i)$$

$$= \sum\nolimits_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log p(c_{w'}|c_w) \cdot p(w'|c_{w'})$$

- Hint: We are now summing over bigrams, that's why we need to add the frequencies $\#(\mathrm{w}, w')$

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log p(c_{w'}|c_w) \cdot p(w'|c_{w'})$$

$$= \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'})}{\#(c_w)} \cdot \frac{\#(w', c_{w'})}{\#(c_{w'})}$$

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'})}{\#(c_w)} \cdot \frac{\#(w', c_{w'})}{\#(c_{w'})}$$

$$= \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'})}{\#(c_w)} \cdot \frac{\#(w')}{\#(c_{w'})}$$

- Hint: Holds, since exactly one cluster per word-type

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'})}{\#(c_w)} \cdot \frac{\#(w')}{\#(c_{w'})}$$

$$= \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'})}{\#(c_w)} \cdot \frac{\#(w')}{\#(c_{w'})} \cdot \frac{n}{n}$$

- Hint: m is the amount of observed bigrams, n is the amount of tokens $n = m + 1$ (think of a sliding window)

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'})}{\#(c_w)} \cdot \frac{\#(w')}{\#(c_{w'})} \cdot \frac{n}{n}$$

$$= \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'}) \cdot n}{\#(c_w) \cdot \#(c_{w'})} \cdot \frac{\#(w')}{n}$$

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'}) \cdot n}{\#(c_w) \cdot \#(c_{w'})} \cdot \frac{\#(w')}{n}$$

$$= \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'}) \cdot n}{\#(c_w) \cdot \#(c_{w'})} + \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(w')}{n}$$

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'}) \cdot n}{\#(c_w) \cdot \#(c_{w'})} + \sum_{w,w'} \frac{\#(\mathrm{w}, w')}{\mathrm{m}} log \frac{\#(w')}{n}$$

$$= \sum_{c,c'} \frac{\#(\mathrm{c}, c')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'}) \cdot n}{\#(c_w) \cdot \#(c_{w'})} + \sum_{w'} \frac{\#(w')}{\mathrm{m}} log \frac{\#(w')}{n}$$

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \sum_{c,c'} \frac{\#(\mathrm{c}, c')}{\mathrm{m}} \log \frac{\#(c_w, c_{w'}) \cdot n}{\#(c_w) \cdot \#(c_{w'})} + \sum_{w'} \frac{\#(w')}{\mathrm{m}} log \frac{\#(w')}{n}$$

$$= \sum_{c,c'} \mathrm{p(c, c')log} \frac{\mathrm{p(c, c')}}{\mathrm{p(c) \cdot p(c')}} + \sum_{w} p(w) log\ p(w)$$

- Note: this does only hold if we assume $\mathrm{n}, m \rightarrow \infty$

Brown Clustering – The model

- As objective function, we use a function $Quality(C)$, which calculates a quality for a given clustering:

$$Quality(C) = \underbrace{\sum_{c,c'} \mathrm{p}(\mathrm{c},\mathrm{c}')\log\frac{\mathrm{p}(\mathrm{c},\mathrm{c}')}{\mathrm{p}(\mathrm{c})\cdot\mathrm{p}(\mathrm{c}')}}_{\boxed{\text{Mutual Information between neighbouring clusters}}} + \underbrace{\sum_{w} p(w) log\ p(w)}_{\boxed{\text{Word-Entropy H (constant!)}}}$$

Brown Clustering-Algorithm-Revisited

- The iterative approach:
  1. Initially assign a distinct cluster to every word
  2. Merge the two cluster that maximize the following objective function:

$$Quality(C) = \sum_{c,c'} \mathrm{p}(\mathrm{c},\mathrm{c}')\log\frac{\mathrm{p}(\mathrm{c},\mathrm{c}')}{\mathrm{p}(\mathrm{c})\cdot\mathrm{p}(\mathrm{c}')} + \sum_{w} p(w)log\ p(w)$$

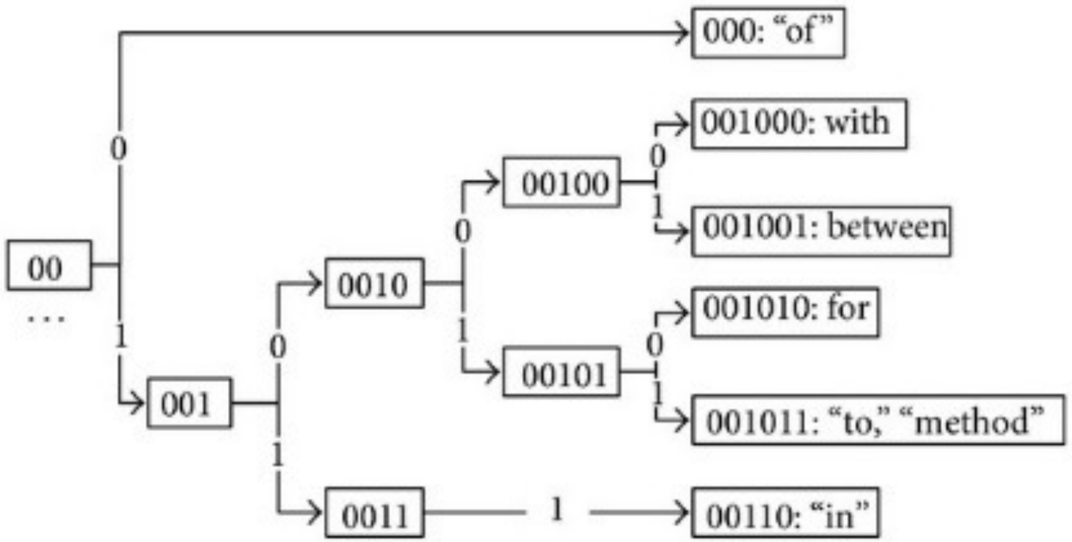
  3. Repeat step 2 until only a single cluster left

- Runtime : For k clusters we need $O(n-k)$ iterations. Each iteration has to evaluate all k² possible merges, each having the cost of $O(k^2)$

➔ Runtime is $O(k^5)$ (but can be speed up to $O(k^3)$)

Brown Clustering-Algorithm

- The algorithm has a nice side-effect, since it produces a binary tree



➔We can now derive nice fingerprints for our words!

Corresponding to different levels in the binary tree