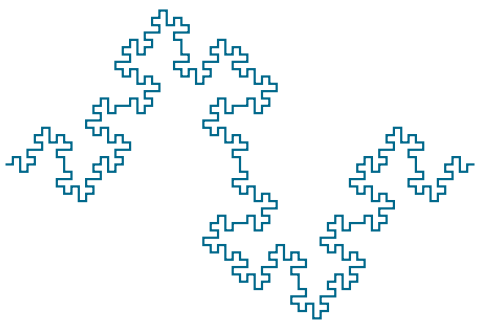
Identifying Disease related Wikipedia Articles

Shashank Shekhar



November 8, 2015

MODEL

A Wikipedia HTML file is modeled as Article. An Article is made of these components :

- sentence (s) - A set of unique words from a tokenized list of text. A sentence inherits its category from its article.
- title (t) - A sentence constructed by tokenize-ing the page heading with white space delimiter.
- content-type (ct) - A sequence of sentence where each sentence is a section heading from the toc section of the html.
- introduction (txt)- A sequence of sentences. constructed by first sentence segmentation and then tokenizing the text available between " tags in the html.
- category - The label associated with this article can be Disease, NotDisease or Undefined

NAIVE BAYES CLASSIFIER

The probability of an article a with title t, content-Type ct and Introduction txt, belonging to class c is calculated as :
$P(a \mid c) = P(t, ct, txt \mid c)$
Three assumptions are made:

- The three features (t, ct, txt) are independent of each other
  $P(a \mid c) = P(t \mid c) * P(ct \mid c) * P(txt \mid c)$
- Each feature is a set of sentences S_g of size g_s where g can be (t, ct, txt)
  $P(g \mid c) = \prod_{s \epsilon g_s} P(s \mid c)$
- A sentence s is a bag of unique words, of size s_w
  $s = (e_1, e_2, ...e_i, e_{s_w})$
  $P(s \mid c) = \prod_{i=1}^{s_w} P(e_i \mid c)$
- A word e in a sentence s is a Bernoulli RV
  $e = 1_{e \epsilon s}$. Here 1 is an indicator function.

TRAINING

A collection of sentences $S^c_{g_all}$ is collected for each feature g across all articles in training set TS. Naive Bayes classifier is trained for each $S^c_{g_all}$.
Here c is the category and $g_{all} = \cup$ (g in TS)

$$\overline{P}(e_i = e \mid c) = \sum_{s \epsilon S^c_{g_{all}}} \sum_{l=1}^{s_w} 1_{e_l = e} / \sum_{s \epsilon S^c_{g_{all}}} s_w$$

CLASSIFYING

The classifier assigns a category c_{MAP} with maxiumum likelihood to an Article a with title t, content-Type ct and Introduction txt:

$$c_{MAP} = argmax_{c \epsilon (Disease, NotDisease)} log_e(P(a \mid c)) + log_e(P(c))$$

- $P(a \mid c) = P(t \mid c) * P(ct \mid c) * P(txt \mid c)$
- $P(t \mid c) = P(s \mid c)$, S_t^c is always of size one
- $P(ct \mid c) = \overline{P}(s \mid c)$ is the mean value across all sentences $s \epsilon S_{ct}^c$
- $P(ct \mid c) = \overline{P}(s \mid c)$ is the mean value across all sentences $s \epsilon S_{txt}^c$

EDGE CASE : DISEASE GROUP

An Article a' with introduction txt' is declared a disease group if :

$$E[\sum_{s \epsilon S_{txt}^{Disease}} \sum_{w \epsilon s} 1_{w=\text{"}diseases\text{"}}] * factor < \sum_{s \epsilon txt'} \sum_{w \epsilon s} 1_{w=\text{"}diseases\text{"}}$$

Some Success Cases:

- Mitochondrial disease
- Sexually transmitted infection
- Transmissible spongiform encephalopathy
- Cancer classified as Disease for factor 5 :(However for factor 3 it is not and also Typhus is not but accuracy drops to 0.980.

EDGE CASE :CHEMICAL SUBSTANCE

An Article a' is declared a Chemical Substance if :
The infobox in its HTML has a *CAS Registry Number*

ERROR ANALYSIS

Training Error

Articles labeled as Disease = 3692

Articles labeled as NotDisease = 10000

- Accuracy = 0.982

Cross Validation Error

Hold Out Size = 500

- Accuracy = 0.958

Training Error with Edge-Case Handling

- Accuracy = 0.983

INSTRUCTIONS TO BUILD

1. **Install sbt (http://www.scala-sbt.org/download.html)**
2. ‘cd‘ into the directory **wikiclassifier**
3. ‘mkdir output‘
4. ‘mkdir training‘. In **training** directory ‘mkdir positive‘ and ‘mkdir negative‘. Then copy all labeled html files with label Disease into **positive** and rest in **negative**.

INSTRUCTIONS TO RUN

1. **Train and find Training Error**
sbt 'runMain org.shkr.wikiclassifier.Training train'

2. **Train and run Cross-Validation with hold-out size** 500
sbt 'runMain org.shkr.wikiclassifier.Training cv 500'

Both 1 & 2 print error table and write additional output to 'output/error_analysis.txt'.

3. **Train then classify wikipedia pages using urls**
en.wikipedia.org/wiki/Cancer, *en.wikipedia.org/wiki/Baseball*
sbt 'runMain org.shkr.wikiclassifier.Training classify Cancer,Baseball

It writes a file for each url in 'output/wikiname_label.json'.
The json has extracted information and the label is Disease or NotDisease

4. **Add flag** '--edgecase' to enable edge case handling for any command, example :
sbt 'runMain org.shkr.wikiclassifier.Training train --edgecase'