Text Classification
Lab 3 - SEng 474 / CSC 578D
Data Mining

Yudi Santoso

Introduction

- A typical task when working with documents is to classify them according to some category.
- In this information era, the number of documents is overwhelming. How can we automatise the task?
- But first, how do we classify documents?
- Let us consider the following two texts:

Example

(A): “A computer mouse is a pointing device (hand control) that detects two-dimensional motion relative to a surface. This motion is typically translated into the motion of a pointer on a display, which allows a smooth control of the graphical user interface.” [Wikipedia]

(B): “Domestic mice sold as pets often differ substantially in size from the common house mouse. This is attributable both to breeding and to different conditions in the wild. The most well known strain, the white lab mouse, has more uniform traits that are appropriate to its use in research.” [Wikipedia]

Introduction

- We would know right away that (A) is related to computers, while (B) is related to a type of rodent.
- But, how do we tell?
  - Context.
    - Natural to human, but not (yet) to computer.
  - Words content.
    - Find keywords, frequencies.

Example (A)

"Domestic mice sold as pets often differ substantially in size from the common house mouse. This is attributable both to breeding and to different conditions in the wild. The most well known strain, the white lab mouse, has more uniform traits that are appropriate to its use in research." [Wikipedia]

The terms (lower-cased, sorted alphabetically):

appropriate(1), attributable(1), breeding(1), common(1), conditions(1), differ(2), domestic(1), house(1), known(1), lab(1), mice(1), mouse(2), pet(1), research(1), size(1), sold(1), strain(1), substantially(1), traits(1), uniform(1), use(1), white(1), wild(1)

Example (B)

"A computer mouse is a pointing device (hand control) that detects two-dimensional motion relative to a surface. This motion is typically translated into the motion of a pointer on a display, which allows a smooth control of the graphical user interface." [Wikipedia]

The terms (lower-cased, sorted alphabetically):

allow(1), computer(1), control(2), detects(1), device(1), dimensional(1), display(1), graphical(1), hand(1), interface(1), motion(3), mouse(1), pointing(1), pointer(1), relative(1), smooth(1), surface(1), translated(1), two(1), typically(1), user(1)

Building Text Classifier (1)

- We have ignored words that (we think) should not affect the category:
  - and, are, as, both, from, has, in, is, more, most, often, the, this, etc.
  - These are called stopwords. Note that sometimes they depend on the category (e.g., plural vs singular).
- There are other processings as well, like finding word root (e.g., graphical -> graph), etc.

Building Text Classifier (2)

- Suppose we want to classify texts as *related* or *not related to rodent*. Now, which words are the keywords? Which ones are the most important? How do we build the rules for the classifier? Will it generalise for new texts?
- Machine Learning:

  Let the computer build the model by itself.

  - Just give examples (i.e., training datasets).
  - And the learning algorithm (e.g., Naive Bayes - discussed in class).

Text Classification with Weka

Weka can do text classification too. It has some learning algorithms that can be used for text classification, e.g., Naive Bayes. It also provides example datasets:

- `ReutersCorn`
- `ReutersGrain`

Both with separate train and test sets. Let us see how we do it, but first let us explore the datasets.

Exploring the Dataset

- Open Weka Explorer
- Open File - Choose `ReutersCorn-train.arff`
- Notice that there are only two attributes: text, and class-att (0,1).
- We can also check this dataset with a text editor (e.g., Notepad++).
- To change the text string to word items, use **Filter** - `Choose > filters > unsupervised > attribute > StringToWordVector`

Classify (1)

- Click `Undo` - back to text attribute.
- Click `Classify` tab.
- On **Test Options** click `Supplied test set > Set`. Choose `ReutersCorn-test.arff`.
- On **Classifier** click `Choose > classifiers > meta > FilteredClassifier`.
- Click on **FilteredClassifier**. Set the `classifier` to `J48`. Set the filter to `StringToWordVector`. Click `OK`.

Classify (2)

- Click `Start` - let it runs.
- Check the result.
- Click on **FilteredClassifier**. Now set the `classifier` to `NaiveBayes`. Keep the filter set to `StringToWordVector`. Run it.
- Repeat with `NaiveBayesMultinomial`.
- Analyse the results.

Closing

- In this lab, we learn about simple document classification. These topic is within a branch of Data Mining called Automatic Document Classification (ADC). It is also a part of NLP (Natural Language Processing).
- We have learned how to do text classification with Weka, using J48 and NaiveBayes classifiers.
- There are many other topics in ADC and NLP as well. Let’s keep learning !!