Information Extraction

Review of Übung 2

CIS, LMU München

Winter Semester 2016-2017

Dr. Alexander Fraser, CIS

Administravia

- Seminar:
- Hausarbeit is due 3 weeks after your presentation
- However, Xmas break (24th to 6th) does not count towards your three weeks
  - Add two more weeks if your working period touches these dates

- I will present a quick review of the Übung to make sure you have the key concepts
- If you are one of the few people who are not in the Seminar
  - You will still be able to follow what I am discussing
  - You can try doing the Übung (Exercise2) by simply going to the Seminar web page and downloading the relevant materials

Review of Übung

- In the Übung last week, we used the open source machine learning package Wapiti
- We worked on a binary learning task: finding <stime> tags
- We looked at:
  - Basic setup (compiling Wapiti, create sa-tagged directory) – "make prep"
  - How to run experiments (train, development, test) – "make"
  - Basic feature extraction code "extract_003.pl"
  - Wapiti pattern files "unigram_bigram_pattern.txt"

How to run experiments (train, development, test)

- Ideally you should run shell scripts like this:

  bash myscript.sh >& myscript.sh.log

- This saves the output into a log file (I always do this, and none of my scripts take parameters)
  - Even better would be to have the extractor print version numbers (and maybe use source control)

Basic feature extraction code

- We looked at extract_003.pl
- This extracts a raw representation which I sometimes refer to as the "features", but which should really be referred to differently
  - Let's call what this outputs the extract file
- The extract file is used to build the actual features used by Wapiti (and contains the gold-standard labels for training data or test data where we want Wapiti to calculate precision/recall and F)

Wapiti pattern files

- Wapiti pattern files are a level of indirection that allow us to:

  1) specify whether a column in the extract file is used
  - This is useful to "comment out" features in the extract file
  - Otherwise it is annoying – you have to remember to explicitly enable each new column as a feature

  2) create features that combine columns (so-called "compound" features)
  - Two features put together is often called a bigram

Beyond binary classification

- Wapiti supports multi-class classification
- You can just change the label in the last column in the "extract" file to any string
- Then retrain
- Very abstractly, it is doing something like one-against-all as I explained in class
  - The details are more complicated, in fact it is a multi-class maximum entropy model
  - I will skip the details (at least for now)

Sequence classification

- There is also a script that does sequence classification
- When using sequence classification, you have several rows like in the extract file
  - But without blank lines between them
  - This is a sequence
- You define a special feature which says "look at the previous label" (this feature starts with the letter "b" in the Wapiti pattern file, because it is defining a feature on the previous label and current label, which is a *label* bigram feature)
- You'll notice that the extract is much simpler, because we can refer to the word in the previous example, or the word in the next example (instead of including these as columns as we did previously)
- We will look at sequence classification in a further lab after the break

Conclusion

- Wapiti is a very interesting package for multi-class and sequential multi-class classification
- It is also quite easy to use
  - Except the annoying bug that we engineered around (where we added a single letter to very simple features like "isUpper" or "isNotUpper")
- Read the manual to see what it can do
- A further detail for avoiding overfitting the training corpus is a technique called "regularization"
  - See the Wapiti paper (cited on the website) for more about this

- Thank you for your attention!