SIMPLY SCALA

EUROCLEAR TM HACKATHON

APPROACH

SIMPLY SCALA

- Considered most of the label identification as a classification problem
- Except for amounts: regular expression matching
- first tried NER with libraries from Stanford
  - not satisfactory
  - slow
- classification with Stanford classifier
  - very good results on the first train/test set
  - with training of 7000 docs: Out of Memory error (memory leaks ?)
- classification with Spark:
  - cluster on Amazon EC2 with master and 4 slaves (6GB, dual core)
  - training using Spark ML pipeline, 3 fold cross validation
  - hyper parameter grids for tuning

PRE-PROCESSING

SIMPLY SCALA

- check OCR'ed text file if they're valid: count the number of non-word characters
- if not valid:
  - remove alpha layer from pdf
  - convert to tiff
  - OCR with tesseract
- Input for classifiers:
  - use whole document
  - minimal preprocessing for Stanford classifier
  - stop word removal for Spark jobs

ALGOS

SIMPLY SCALA

- Stanford classifier:
  - maximum entropy (softmax) classifier, similar to multi class logistic regression
  - has been worked on for over a decade
  - intimate knowledge of the English language
  - for multi-label (ROCs) I used ‘problem transformation’ (binary classifiers for each label/ROC)
- Spark:
  - feature extraction with word2vec
  - logistic regression for binary classifiers (zero coupon)
  - experimented with “multilayer perceptron” (neural network)
    - too time consuming
  - used random forest for multi class

CHALLENGES

SIMPLY SCALA

- wasted a lot of time trying to combine input files
- memory leak problem with Stanford classifier
- machine learning in Spark is not as mature as I expected:
  - many classifiers are only binary: logistic regression, gradient boosted trees
  - impossible to save state of trained classifiers
  - on the other hand: provides an elegant pipeline interface, inspired by Python

FUTURE IMPROVEMENTS

SIMPLY SCALA

- Focus on content, domain
- Use of generic machine learning for NLP ?
- Get the input organised
- Possible implementation of trained classifiers:
  - deploy in Scala Akka framework
  - each classifier is an actor, working in parallel
  - drag and drop pdf on a web page
  - receive answers from actors through web sockets