Kaggle WISE2014. 2nd-place Solution

Team anttip: Antti Puurula[1] and Jesse Read[2]

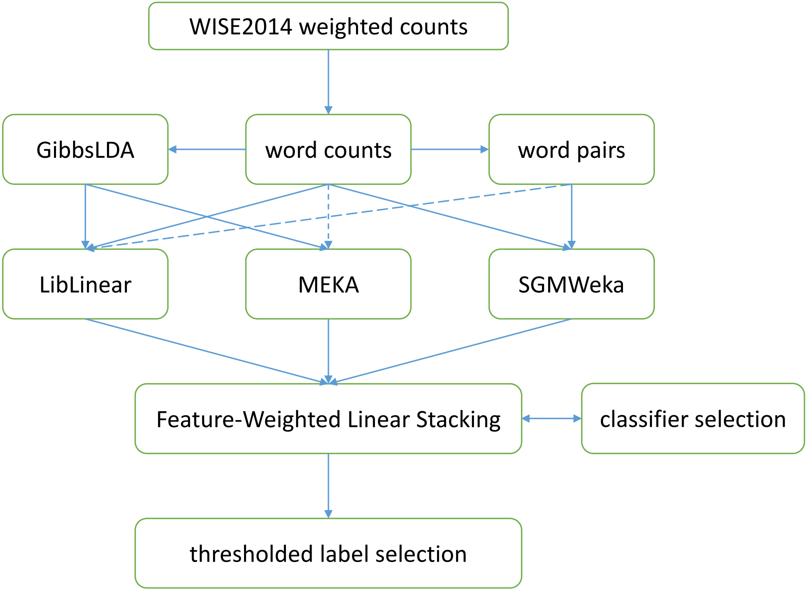
(1) University of Waikato, New Zealand
(2) Aalto University and HIIT, Finland

12 October 2014

Overview

An ensemble of diversified base-classifiers combined with a variant Feature-Weighted Linear Stacking

- Features:
  - Word counts, LDA features, word pair features
  - TF-IDF and other optimized transforms applied to some features
- Base-classifiers:
  - Extensions of MNB and Multinomial Kernel Density models
  - Logistic Regression, SVM, and tree-based classifiers
- Problem-transformation methods:
  - Binary relevance, classifier chains, and label-powerset based methods (incl. pruned sets and RAkEL)
- Ensemble:
  - Feature-Weighted Linear Stacking with hill-climbing classifier selection
  - Thresholded label selection from the top label candidates


WISE2014 weighted counts
GibbsLDA
word counts
word pairs
LibLinear
MEKA
SGMWeka
Feature-Weighted Linear Stacking
classifier selection
thresholded label selection

Data Segmentation

Documents	Used as
1—58857	training base classifiers
next 5000	5×1000 for base-classifier optimization
final 5000	ensemble learning set

Features

- **Original word counts** recovered using a reverse TF-IDF search:
  - reverse the IDF and log-transforms, constrain the minimum count of a word to 1, and solve for the missing document length norm variable
- **Topic features** with Gibbs LDA++
  - computed 5 different topic decompositions (ranging from 50—300 topics per document) with parameters and pre-processing choices recommended in the literature
- **Word pair** features:
  - use IDF and count thresholds to prune possible pairs, represent each document with pruned word pairs
  - total 6011508 pruned word pairs, mean 227.33 per document
- Features further transformed with TF-IDFs depending on the classifier

Problem Transformation

- Multi-label problem transformation methods
  - binary relevance (BR)
  - classifier chains (CC)
  - label powerset (LP)
  - pruned label poweset (PS)
  - random [pruned] labelsets (i.e., RAkEL+PS)
  - chained random labelsets (i.e., CC+RAkEL)

Summary of Toolkits Used

Base classifiers	Toolkit	Prob. transform.	Features
gen., algebraic	`SGMWeka`	LP, PS	words, word pairs
discriminative	`LibLinear`	BR, CC, RAkEL	words, LDA
discriminative	`Meka`	RAkEL, PS, CC	LDA, words

- In `SGMWeka` and `LibLinear`, base classifiers were optimized using 40x20 Gaussian Random Searches (Puurula 2012) on the 5x1000 development folds.
- In `Meka`, parameters for base classifiers were chosen randomly upon each instantiation, from sensible ranges
- Heavy pruning and small subsets in some cases, particularly for tree-based methods

SGMWeka

- Generative (MNB, ...) and algebraic (Centroid, ...) models
- Extensions of MNB such as *Tied Document Mixture* (Puurula & Myang 2013)
  - Hierarchical smoothing with Pitman-Yor Process LM, Jelinek-Mercer
  - Model-based feedback
  - Exclusive training subsets for the ensemble
- Label powerset methods most scalable in this framework

See `https://sourceforge.net/projects/sgmweka/` for details.

LibLinear

- Discriminative classifiers (SVM, LR) with L1 regularization worked best
- Words and LDA used (word pairs didn't work well)
- Used binary relevance and classifier chains transformations (label powerset methods were not scalable)
- Also tried: Chained Random Labelsets (CC becomes more scalable this way)

Meka

Meka classifiers (≈ 100) with randomly chosen ...

feature space	one of the five LDA transforms
base classifier (Weka)	one of SMO, J48, SGD
... and parameters	e.g., -C for SMO, pruning for trees
problem trans. (Meka)	RAkEL-PS, *RAkELd-PS*, PS, or CC-RAkEL
... and parameters	m sets of k labels, with p, n pruning
feature subspace	5 to 80 percent
instance subspace	5 to 80 percent

- also tried with original words feature space, but quite slow

See meka.sourceforge.net for details.

Ensemble: Feature-Weighted Linear Stacking

- Approximate optimal weights for each instance and classifier using an oracle
- Predict vote weight of each base-classifier using meta-features:
  - document L0-norm
  - output labelset properties (e.g., frequency in training set)
  - output labelset for neighbouring documents
  - correlation of the labelsets to predictions of other base classifiers
- Features transformed by ReLU and log-transforms
- Use a Random Forest for each base classifier and its meta-feature set

Ensemble: Threshold Selection

- Sum a score for each label, and
- Threshold on the maximum score for the document, such that labels with score $>$ 0.5 * max_score are selected

Ensemble: Base-classifier Selection

- Select base classifiers to optimize ensemble Mean F-score performance
- Parallelized hill-climbing Tabu-search
    - steps of addition, removal or replacement of a base-classifier
    - random restarts
    - penalization term on the number of base-classifiers (accelerated optimization considerably)
- Final ensemble:
    - around 50 base-classifiers, from over 200 generated

Discussion / Lessons Learned

- Data segmentation is critical, leave the last training set documents for optimization
  - reduces overfitting
- L1-regularized linear base-classifiers worked best
  - we should have used data weighting and label-dependent parameters
- Scalability becomes an issue for problem transformation with `Weka`-based frameworks
  - the `Instance` class is a bottleneck: attribute space copied many times internally
  - can train base-classifiers one-at-a-time, or use heavy subsampling
- Ensemble combination saved the day:
  - our base-classifiers scored lower than other teams, but were very diverse

The End

Thank you for your attention.

- Antti Puurula: http://www.cs.waikato.ac.nz/~asp12/
- Jesse Read: http://users.ics.aalto.fi/jesse/