Kaggle WISE2014. 2nd-place Solution

Team anttip: Antti Puurula\textsuperscript{(1)} and Jesse Read\textsuperscript{(2)}

\textsuperscript{(1)} University of Waikato, New Zealand
\textsuperscript{(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

LibLinear

MEKA

SGMWeka

Feature-Weighted Linear Stacking

classifier selection

thresholded label selection

Antti Puurula and Jesse Read

Kaggle WISE2014. 2nd-place Solution
<table>
<thead>
<tr>
<th>Documents</th>
<th>Used as</th>
</tr>
</thead>
<tbody>
<tr>
<td>1—58857</td>
<td>training base classifiers</td>
</tr>
<tr>
<td>next 5000</td>
<td>$5 \times 1000$ for base-classifier optimization</td>
</tr>
<tr>
<td>final 5000</td>
<td>ensemble learning set</td>
</tr>
</tbody>
</table>
**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
Multi-label problem transformation methods

- binary relevance (BR)
- classifier chains (CC)
- label powerset (LP)
- pruned label powerset (PS)
- random [pruned] labelsets (i.e., RAKEL+PS)
- chained random labelsets (i.e., CC+RAkEL)
### Summary of Toolkits Used

<table>
<thead>
<tr>
<th>Base classifiers</th>
<th>Toolkit</th>
<th>Prob. transform.</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>gen., algebraic</td>
<td>SGMWeka</td>
<td>LP, PS</td>
<td>words, word pairs</td>
</tr>
<tr>
<td>discriminative</td>
<td>LibLinear</td>
<td>BR, CC, RAKEL</td>
<td>words, LDA</td>
</tr>
<tr>
<td>discriminative</td>
<td>Meka</td>
<td>RAKEL, PS, CC</td>
<td>LDA, words</td>
</tr>
</tbody>
</table>

- 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.
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.
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 classifiers ($\approx 100$) with randomly chosen ...  

<table>
<thead>
<tr>
<th>Feature Space</th>
<th>One of the five LDA transforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Classifier (Weka)</td>
<td>One of SMO, J48, SGD</td>
</tr>
<tr>
<td>... and Parameters</td>
<td>E.g., -C for SMO, pruning for trees</td>
</tr>
<tr>
<td>Problem Trans. (Meka)</td>
<td>RAKEL-PS, RAKELd-PS, PS, or CC-RAkEL</td>
</tr>
<tr>
<td>... and Parameters</td>
<td>$m$ sets of $k$ labels, with $p$, $n$ pruning</td>
</tr>
<tr>
<td>Feature Subspace</td>
<td>5 to 80 percent</td>
</tr>
<tr>
<td>Instance Subspace</td>
<td>5 to 80 percent</td>
</tr>
</tbody>
</table>

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

See meka.sourceforge.net for details.
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 $> 0.5 \times \text{max score}$ are selected
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
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
Thank you for your attention.

- Jesse Read: http://users.ics.aalto.fi/jesse/