Learning Word-to-Concept Mappings for Automatic Text Classification

Georgiana Ifrim  Martin Theobald
Gerhard Weikum
Outline

- Introduction
- Related Work
- Probabilistic Model
- Experimental Results
- Conclusion & Future Work
Introduction

Motivation

- Richness of features to represent documents – possible bottleneck for obtaining high accuracy in text categorization
- Try to exploit language semantics to overcome it

Our Contribution

- Probabilistic mapping of words to their meanings through an iterative EM process, coupled with a classifier for topic labelling
- Bootstrapping initialization heuristic for avoiding combinatorial explosion of parameter space and possible EM flaws
Related Work

- **Work on spectral decomposition**
  - Deerwester, Dumais & Harshman, 1990 (LSA)
  - Hoffman, 2001 (PLSA)
  - Represent documents in a reduced space of concepts; Require specifying number of concepts/actors *a priori*

- **Feature engineering & Wordnet (WN)**
  - Cai & Hoffman, 2003 (PLSA + AdaBoost)
  - Bloedhorn & Hotho, 2004 (explicit concepts from WN + AdaBoost)
  - Scott & Matwin, 1998 (explicit concepts from WN + RIPPER)
  - Enhance word feature space by concepts & plug into classifier
Problem Setup

Given

- A data collection (Reuters-21578, Amazon)
  - A set of training documents with known topic labels and observed features, but latent concepts
- An ontology of concepts (WordNet)
  - Each concept has a set of synonyms, a short textual description and is linked by hierarchical relations

Goal

- For a new test document, predict its topic label
Relate features to topics through latent concepts
Probabilistic Model (II)

- Generative process for feature-topic pairs

- Select a topic $t$ with probability $P[t]$
- Pick a latent variable $c$ with probability $P[c|t]$ (Probability that concept $c$ describes topic $t$)
- Generate a feature $f$ with probability $P[f|c]$ (Probability that word $f$ means concept $c$)
Probabilistic Model (III)

- Associate with each observation (feature f, topic t) a latent variable (concept c)

\[ P[f, t] = \sum_{c \in C} P[c] \cdot P((f, t) | c) \]

- Independence assumptions:
  - Observation pairs (f, t) are generated independently
  - Conditioned on the latent variable c, features f are generated independently of topic t

- Log-likelihood of the observed pairs (f, t):

\[
\log L = \log \left( \prod_{(f, t)} P[f, t]^{n(f, t)} \right) \\
l = \sum_{(f, t)} n(f, t) \cdot \log(P[f, t]) = \sum_{(f, t)} n(f, t) \cdot \log(\sum_{c \in C} P[c] \cdot P((f, t) | c))
\]
Estimate model parameters \{P[t], P[f|c], P[c|t]\} so as to maximize the complete log-likelihood of (f,t) pairs (EM algorithm):

\[
E[l^{\text{comp}}] = \sum_t \sum_f n(f, t) \cdot \sum_{c \in C} P[c | (f, t)] \cdot \log(P[t] \cdot P[f | c] \cdot P[c | t])
\]

Use Bayes rule & learned parameters

\[
t = \arg\max_t P[t | d] = \arg\max_t P[d | t] \cdot P[t] = \arg\max_t \prod_f P[f, t]
\]

\[
(P[d | t]) \cdot P[t] = (\prod_{f \in d} P[f | t]) \cdot P[t] = \prod_{f \in d} P[f, t]
\]

\[
P[f, t] = \sum_c P[t] \cdot P[f | c] \cdot P[c | t]
\]
Problems with EM (I)

1. Large number of model parameters

Solution: Prune the parameter space

- Feature selection (Mutual Information)
  - Extract phrases, exploit PoS information

- Concept selection (Ontology)
  - Select a subset of concepts from the ontology, that reflects well the semantics of the given training collection
  - For a given feature \( f \), extract all meanings
  - Refine this ‘mapping’ by EM learning

⇒ Reduces computational complexity, Increases model robustness
Problems with EM (II)

2. Risk of local maxima

Solution: Pre-initialize model parameters

- Context based similarity => probability
  - Context(f) = text window in document
  - Context(c) = hypernyms, hyponyms, siblings + their glosses from ontology
  - Context(t) = top features selected by MI from training collection of topic t

\[
P[f | c] = \frac{\text{sim}(\text{context}(f), \text{context}(c))}{\sum_{f \in F} \text{sim}(\text{context}(f), \text{context}(c))}, \quad (\sum_{f \in F} P[f | c] = 1, \ \forall c \in C)
\]

\[
P[c | t] = \frac{\text{sim}(\text{context}(c), \text{context}(t))}{\sum_{c \in C} \text{sim}(\text{context}(c), \text{context}(t))}, \quad (\sum_{c \in C} P[c | t] = 1, \ \forall t \in T)
\]

⇒ Speeds up convergence, Reduces risk of getting stuck in local max
Experimental Results (I)

- **Reuters-21578**
  - Select top 5 topics: earn, acq, crude, trade, money-fx
  - Training: 1,000 documents; Test: 2,000 documents

"Crude oil prices rallied today, moving over 17.00 dlrs a barrel because of Saudi Arabia's determined effort to support prices, analysts said."

<table>
<thead>
<tr>
<th>Training per topic</th>
<th>Microavg F1 NBayes</th>
<th>Microavg F1 LatentM</th>
<th>Microavg F1 LatentMPoS</th>
<th>Microavg F1 SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>88.9%</td>
<td>88.7%</td>
<td>87.8%</td>
<td>90.0%</td>
</tr>
<tr>
<td>20</td>
<td>89.6%</td>
<td>92.2%</td>
<td>90.7%</td>
<td>92.1%</td>
</tr>
<tr>
<td>30</td>
<td>92.7%</td>
<td>94.0%</td>
<td>92.2%</td>
<td>93.6%</td>
</tr>
<tr>
<td>40</td>
<td>92.1%</td>
<td>93.0%</td>
<td>91.2%</td>
<td>94.5%</td>
</tr>
<tr>
<td>50</td>
<td>93.8%</td>
<td>95.0%</td>
<td>93.8%</td>
<td>93.8%</td>
</tr>
<tr>
<td>100</td>
<td>95.3%</td>
<td>94.9%</td>
<td>93.8%</td>
<td>95.5%</td>
</tr>
<tr>
<td>150</td>
<td>96.0%</td>
<td>95.0%</td>
<td>94.4%</td>
<td>95.4%</td>
</tr>
<tr>
<td>200</td>
<td>95.9%</td>
<td>95.8%</td>
<td>94.5%</td>
<td>95.9%</td>
</tr>
</tbody>
</table>
Experimental Results (II)

Amazon

- Books' editorial reviews from amazon.com
- Select 3 topics: Biological Sciences, Mathematics, Physics
- Training: 1,500 documents; Test: 4,500 documents
- Study vocabulary size (left) & training size influence (right) on model performance

In a place where art, science and technology meet, Joseph Scheer's images of moths emerge. These ubiquitous creatures are often considered drab-colored poor relations of the "beautiful" butterfly;
Experimental results (III)

Amazon

- Similarity based heuristic vs. random initialization of parameters
  - Heuristic speeds-up convergence

- Heuristic vs. Heuristic + 1 EM iteration
  - Neither technique alone can achieve good performance

<table>
<thead>
<tr>
<th>EM ITERATION</th>
<th>SIM-BASED INIT</th>
<th>RANDOM INIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80.5%</td>
<td>59.0%</td>
</tr>
<tr>
<td>2</td>
<td>81.5%</td>
<td>70.6%</td>
</tr>
<tr>
<td>3</td>
<td>81.9%</td>
<td>76.5%</td>
</tr>
<tr>
<td>4</td>
<td>82.2%</td>
<td>79.8%</td>
</tr>
<tr>
<td>5</td>
<td>82.3%</td>
<td>80.9%</td>
</tr>
<tr>
<td>10</td>
<td>82.5%</td>
<td>82.3%</td>
</tr>
<tr>
<td>15</td>
<td>82.5%</td>
<td>82.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TRAINING</th>
<th>HEURISTIC</th>
<th>HEURISTIC-EM1</th>
<th>RANDOM-EM1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>38.1%</td>
<td>56.8%</td>
<td>49.8%</td>
</tr>
<tr>
<td>20</td>
<td>66.6%</td>
<td>60.9%</td>
<td>49.6%</td>
</tr>
<tr>
<td>30</td>
<td>68.2%</td>
<td>67.7%</td>
<td>49.6%</td>
</tr>
<tr>
<td>40</td>
<td>40.3%</td>
<td>70.5%</td>
<td>49.8%</td>
</tr>
<tr>
<td>50</td>
<td>43.4%</td>
<td>71.7%</td>
<td>49.8%</td>
</tr>
<tr>
<td>100</td>
<td>27.3%</td>
<td>74.8%</td>
<td>49.8%</td>
</tr>
<tr>
<td>200</td>
<td>29.9%</td>
<td>79.3%</td>
<td>49.8%</td>
</tr>
<tr>
<td>300</td>
<td>27.6%</td>
<td>80.8%</td>
<td>51.0%</td>
</tr>
<tr>
<td>400</td>
<td>30.4%</td>
<td>80.3%</td>
<td>51.0%</td>
</tr>
<tr>
<td>500</td>
<td>32.3%</td>
<td>80.5%</td>
<td>52.0%</td>
</tr>
</tbody>
</table>
Conclusion & Future Work

- Learning word-to-concept mappings can improve accuracy on certain datasets

- Short-term
  - Experimental work:
    - Semantically richer data collections + Customized ontologies (Wikipedia…suggestions welcome)
  - Different types of classifiers: Bayesian Network

- Long-term
  - Given a data collection, predict if & how much semantics can help
  - Applications of the proposed model in IE, QE
Thank you!