



Transductive Learning for Text Classification using Explicit Knowledge Models

Georgiana Ifrim

Gerhard Weikum

- Introduction
- Related Work
- Transductive Latent Model
- Experimental Results
- Conclusion

□ Motivation

- Many applications require classification models able to learn from little training and background knowledge

- *Amazon, Google, Yahoo!, Wikipedia*

- **Transductive learning**: documents to be labelled are available beforehand (as opposed to Inductive learning)

- Learn from **few labelled docs** and rich **background knowledge**, e.g.:
 - **Unlabeled docs**: learn feature distributions, structure
 - **Explicit knowledge models** (ontologies, e.g. WordNet): learn about concept-word, concept-concept relations, phrases that express concepts
 - **Domain knowledge** (encyclopaedia, e.g. Wikipedia): learn about topic descriptions

□ Contribution

- A latent variable model for transductive text classification which integrates **background knowledge** into the learning process
- Solutions to avoid model selection problems and parameter tuning by using explicit concepts from an ontology and an informative Dirichlet prior distribution on model parameters

❑ Transductive Learning

❑ Bennet99, Joachims99 (TSVM)

- ❑ Adjust maximum margin hyperplane based on both training and test data

❑ Nigam et al.00

- ❑ Use unlabeled data for re-weighting labelled examples

❑ Blum & Chawla01, Joachims03 (SGT)

- ❑ Represent the dataset as a graph and search for mincuts or min average cuts

❑ Latent Models

❑ Deerwester et al.90 (LSA), Hoffman01 (PLSA)

- ❑ Represent documents in a reduced space of concepts/aspects; Require specifying number of concepts/aspects *a priori*

❑ Explicit Knowledge, e.g. WordNet

❑ Bloedhorn & Hotho04, Scott & Matwin98

- ❑ Enhance word feature space by concepts & plug into classifier

□ Given

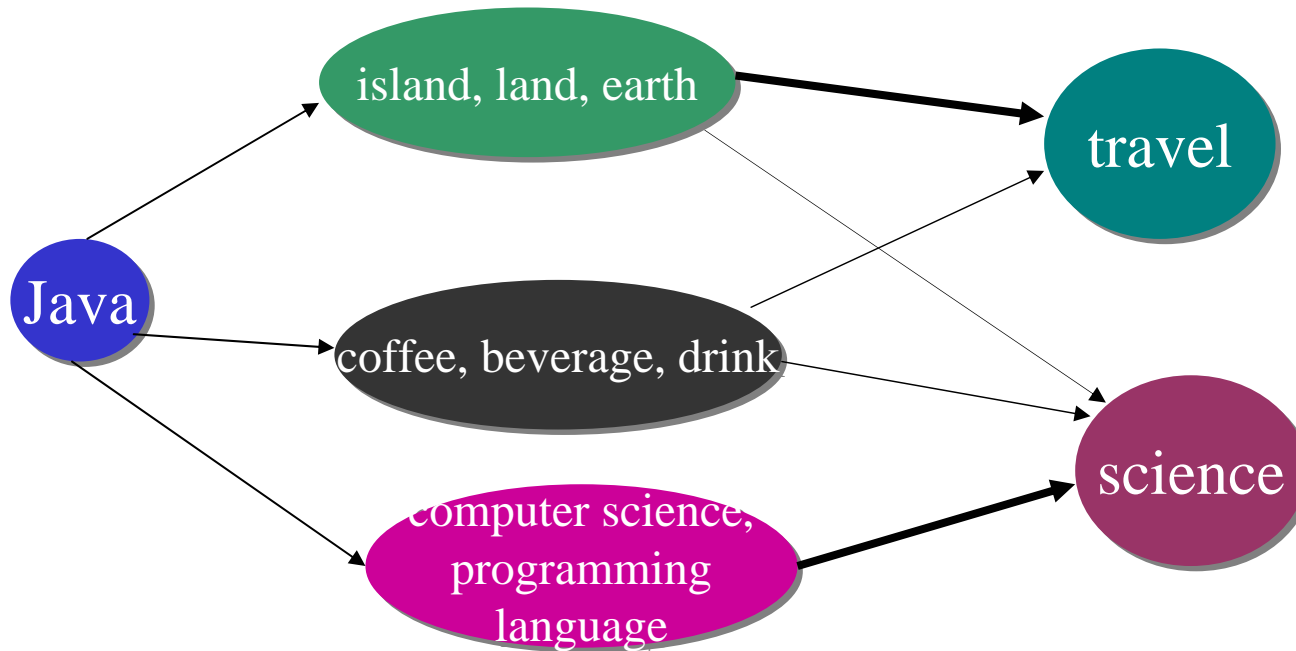
- An unlabeled collection of text documents, split into **small training** (labelled) set and **large test** (unlabeled) set
- Access to an **ontology** of concepts (e.g. WordNet); each concept has a short textual description and is linked to other concepts by semantic edges
- Access to the Web (**encyclopaedia**, e.g. Wikipedia)

□ Goal

- Categorize the test collection into pre-defined topics

- Generative model for feature-topic co-occurrences
 - Associate with each observation (feature f , topic t) a latent variable (concept c)

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



□ Learning Model Parameters

- Given training set and assumption that (f, t) pairs generated from multinomial distribution, write down likelihood of observed (f, t) pairs

$$L = \prod_{(f,t)} P[f, t]^{n(f,t)} \quad P[f, t] = \sum_c P[t] \cdot P[f | c] \cdot P[c | t]$$

- Maximize likelihood to estimate parameters
- Due to assumption of “latent concepts”, direct maximization difficult
- Employ an Expectation-Maximization (EM) algorithm to estimate parameters

$$\Rightarrow P[f|c], P[c|t], P[t]$$

- Classify using Bayes rule & learned parameters

$$t = \operatorname{argmax}_t P[t | d] = \operatorname{argmax}_t P[d | t] \cdot P[t] = \operatorname{argmax}_t \prod_f P[f, t]$$

❑ Problems with EM & Our Solutions

1. Large number of model parameters

❑ Solution: **Prune the parameter space**

- ❑ Feature Selection: control the size of vocabulary (tf-idf)
- ❑ Concept Selection: work with a subset of the ontology relevant for the given text collection
 - ❑ Extract for each feature, the most relevant meaning in the text collection

2. Slow convergence or convergence to local maximum

- ❑ Solution: Pre-initialize parameters with good “guesses”
- ❑ Use **background knowledge**: given (unlabeled) collection, ontology, encyclopaedia

❑ **Slow convergence or convergence to local max**

❑ For each feature f in F do

- ❑ Compute $\text{context}(f)$ in the **given collection**: for each occurrence, the context is a text window centered at the feature
- ❑ Query **WordNet** for possible meanings of f : $C = \{c_1, \dots, c_m\}$
- ❑ For each c in C , compute its $\text{context}(c) = \{\text{synonyms, hypernyms, hyponyms, and their glosses}\}$
- ❑ Compute $\text{sim}(\text{context}(f), \text{context}(c))$ and keep $c = \text{argmax}_c \text{sim}(\text{context}(f), \text{context}(c))$

⇒ Candidate set of concepts & $P[f|c] \sim \text{sim}(\text{context}(f), \text{context}(c))$

❑ For each topic t , query Wikipedia for t

- ❑ $\text{Context}(t) = \{\text{top training terms selected with Mutual Information + retrieved page from Wikipedia}\}$
- ❑ $P[c|t] \sim \text{sim}(\text{context}(c), \text{context}(t))$

- ❑ Enhanced EM Algorithm – Informative Dirichlet Prior
 - ❑ The **similarity-based mapping** of feature-concepts and concepts-topics defines a **prior probability distribution** on model parameters
 - ❑ Bayesian inference tells us how to use this additional information in a **Maximum A Posteriori Estimation** (MAP) of parameters
 - ❑ Postulate Dirichlet prior on model parameters; modify EM to compute MAP estimate, instead of ML estimate

Unlabelled set		Training
Explicit Knowledge		Explicit Knowledge
$P[f c] \sim$ Prior	+	EM-estimate

Domain Knowledge		Training
Explicit Knowledge		Explicit Knowledge
$P[c t] \sim$ Prior	+	EM-estimate

□ Test collections

□ Reuters-21578: news collection

- Top 10 categories, docs labelled with unique topic; 8,024 docs
- Topics: acquisitions, earnings, coffee, crude, interest, money-fx, money-supply, sugar, ship, trade

□ Amazon: books' editorial reviews

- Crawl from amazon.com; 3 topics; 5,634 docs;
- Topics: Biological Sciences, Mathematics, Physics

□ Wikipedia: pages from a topic-focused crawl

- Crawl from wikipedia.org; 7 topics; 5,384 docs
- Topics: Politics, Computer Science, Physics, Chemistry, Biology, Mathematics, Geography,

□ Compared Models

- **Inductive methods** (learn only from training): Multinomial Naïve Bayes, Inductive Latent Model (ILM), Inductive SVM (ISVM)
- **Transductive methods:** Transductive SVM (TSVM), Spectral Graph Transducer (SGT), Transductive Latent Model (TLM)

□ Methodology

- Study how much labelled data is needed for achieving reasonable classification accuracy on the test set
- Split collection randomly into **training** and **test**; vary size of splits, **0.25%** up to **10%**; average over 10 random splits
- Parameters fixed: # EM iterations fixed to 1 (due to our init), # features generative models fixed to 10,000; SVM Light with default parameters and all terms; SGT with parameters suggested in Joachims03 and all terms

□ Reuters21578. Microaveraged F1 for different training set sizes

Training	NB	ILM	ISVM	TSVM	SGT	TLM
0.4% (~30 docs)	64.5	67.1	71.8	57.6	76.5	79.7
0.5%	67.2	70.0	73.5	50.2	79.7	80.7
1% (~80 docs)	77.2	76.5	81.0	60.8	88.6	84.1
2%	83.4	81.7	82.2	72.7	89.9	85.1
5%	91.1	90.0	84.9	87.1	92.1	89.4
10% (~800 docs)	92.9	92.0	88.4	89.4	93.0	89.6

□ Amazon. Microaveraged F1 for different training set sizes

Training	NB	ILM	ISVM	TSVM	SGT	TLM
0.25% (~15 docs)	77.2	72.7	48.7	64.1	77.8	83.7
0.5%	79.1	74.8	63.0	68.0	82.4	84.9
1%	81.1	77.9	72.2	75.6	85.0	85.1
2% (~100 docs)	83.3	81.4	78.2	82.1	86.0	85.2
5%	84.2	83.2	83.8	85.8	87.5	85.5
10% (~500 docs)	85.2	84.9	84.5	86.7	88.1	85.9

□ Wikipedia. Microaveraged F1 for different training set sizes

Training	NB	ILM	ISVM	TSVM	SGT	TLM
0.25% (~15 docs)	72.7	70.4	43.0	32.5	70.5	79.2
0.5%	76.0	74.8	61.3	52.0	77.1	80.5
1%	77.9	76.9	73.2	63.1	79.3	80.9
2% (~100 docs)	80.8	80.7	78.9	69.3	81.4	80.8
5%	83.6	82.1	83.2	77.8	82.9	82.8
10% (~500 docs)	85.8	84.5	85.0	81.6	84.3	84.8

- ❑ Many applications require classification models able to learn from little training and rich background knowledge
- ❑ We propose TLM (Transductive Latent Model), an explicit knowledge model for transductive classification which integrates **background knowledge** into the learning process
- ❑ TLM improves classification accuracy over state of the art methods, when very little training is available

Thank you!