Novel representations and methods in text classification

Manuel Montes, Hugo Jair Escalante

Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico.

http://ccc.inaoep.mx/~mmontesg/

http://ccc.inaoep.mx/~hugojair/

{mmontesg, hugojair}@inaoep.mx

7th Russian Summer School in Information Retrieval
Kazan, Russia, September 2013

Novel representations and methods in text classification

SELF-TRAINING USING THE WEB AS CORPUS

Outline

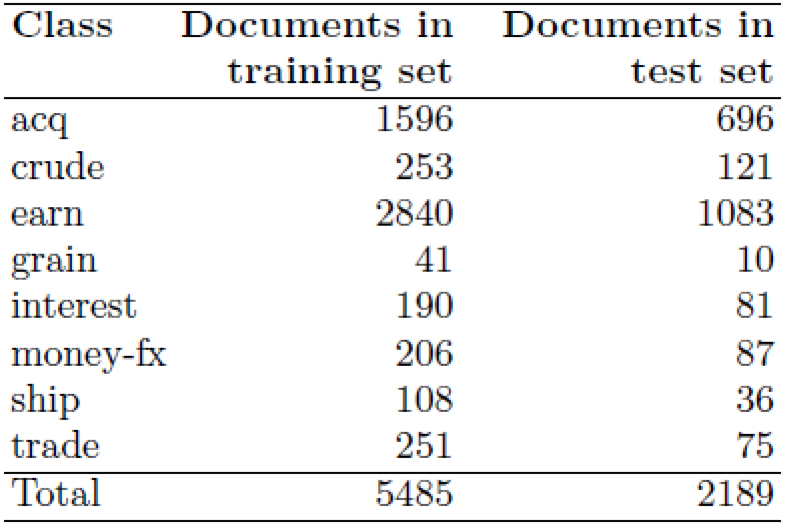
- Issues on supervised text classification
- Introduction to semi-supervised learning
  - Self-training and co-training
- Self-training using the web as corpus
  - Experiments on thematic text classification
  - Experiments on authorship attribution
- Final remarks

Supervised learning

- Most current methods for automatic text categorization are based on supervised learning techniques
- A major difficulty of supervised techniques is that they commonly require large training sets
  - Examples are **manually labeled**
  - Very expensive and time consuming
- Unfortunately, in many real-world applications **training sets are extremely small** and very imbalanced

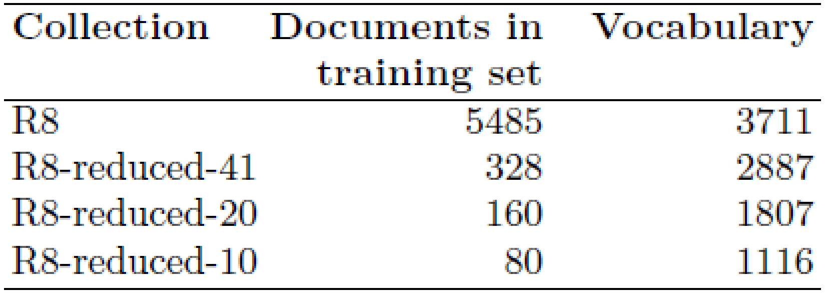
Size of training sets and classification performance

Table 1. The R8 collection



Class	Documents in training set	Documents in test set
acq	1596	696
crude	253	121
earn	2840	1083
grain	41	10
interest	190	81
money-fx	206	87
ship	108	36
trade	251	75
Total	5485	2189

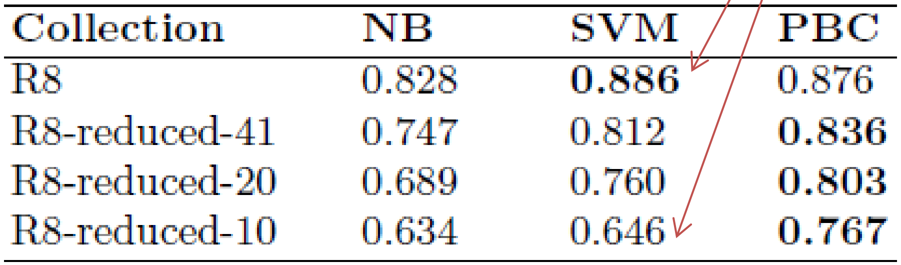
Table 2. The four evaluation datasets



Collection	Documents in training set	Vocabulary
R8	5485	3711
R8-reduced-41	328	2887
R8-reduced-20	160	1807
R8-reduced-10	80	1116

Important drop in accuracy (27%)

Table 3. F-measure results from three classification methods



Collection	NB	SVM	PBC
R8	0.828	**0.886**	0.876
R8-reduced-41	0.747	0.812	**0.836**
R8-reduced-20	0.689	0.760	**0.803**
R8-reduced-10	0.634	0.646	**0.767**

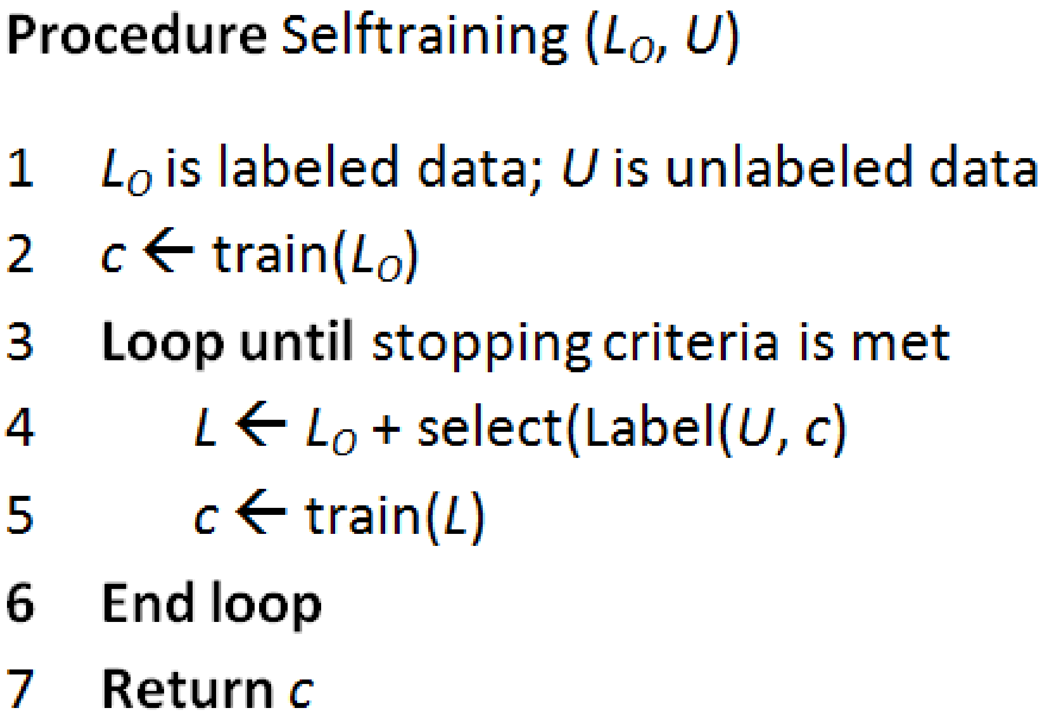
Semi-supervised learning

- Idea: learning from a mixture of labeled and **unlabeled data**.
- This idea was supported on the observation that, for more text classification tasks, it is **easy to obtain samples of unlabeled data.**
- Assumption is that unlabeled data provide information about the joint probability distribution over words and their co-occurrrences

Two main approaches

- **Self training**
  - – Uses its own predictions to **teach itself**
  - – Based on the assumption that “one’s own high confidence predictions are correct”.
- **Co-training**
  - – The idea is to construct **two classifiers** trained on different sub-feature sets, and to have the **classifiers teach each other** by labeling instances where they are able.

Self-training procedure



```
Procedure Selftraining ($L_O$, $U$)

$L_O$ is labeled data; $U$ is unlabeled data
$c$ ← train($L_O$)
Loop until stopping criteria is met
    $L$ ← $L_O$ + select(Label($U$, $c$))
    $c$ ← train($L$)
End loop
Return $c$
```

Parameters and variants

- Base learner: any classifier/ensemble that makes confidence-weighted predictions.
- Stopping criteria: a fixed arbitrary number of iterations or until convergence
- Indelibility: basic version re-labels unlabeled data at every iteration; in a variation, labels from unlabeled data are never recomputed.
- Selection: add only k instances to the training at each iteration.
- Balancing: select the same number of instances for each class, or preserve the initial class proportions.

Co-training procedure

```
Procedure cotraining ($L$, $U$)

$L$ is labeled data, $U$ is unlabeled data
$P$ ← random selection from $U$
Loop until stopping criteria is met
    $F_1$ ← train(view$_1$($L$))
    $F_2$ ← train(view$_2$($L$))
    $L$ ← $L$ + select(label($P$, $F_1$) + select(labelP, $F_2$))
    Remove the labeled instances from $P$ and replenish $P$ from $U$
end loop
```

Comments on semi-supervised methods

- Self-training:
  - The **simplest** semi-supervised learning method, but
  - Early mistakes could reinforce themselves
- Co-training:
  - **Not applicable to all problems**
  - It is necessary to have two different views of the documents.
    - The two features subsets have to be conditionally independent given the class; i.e., high confident data points in one view will be randomly scattered in the other view

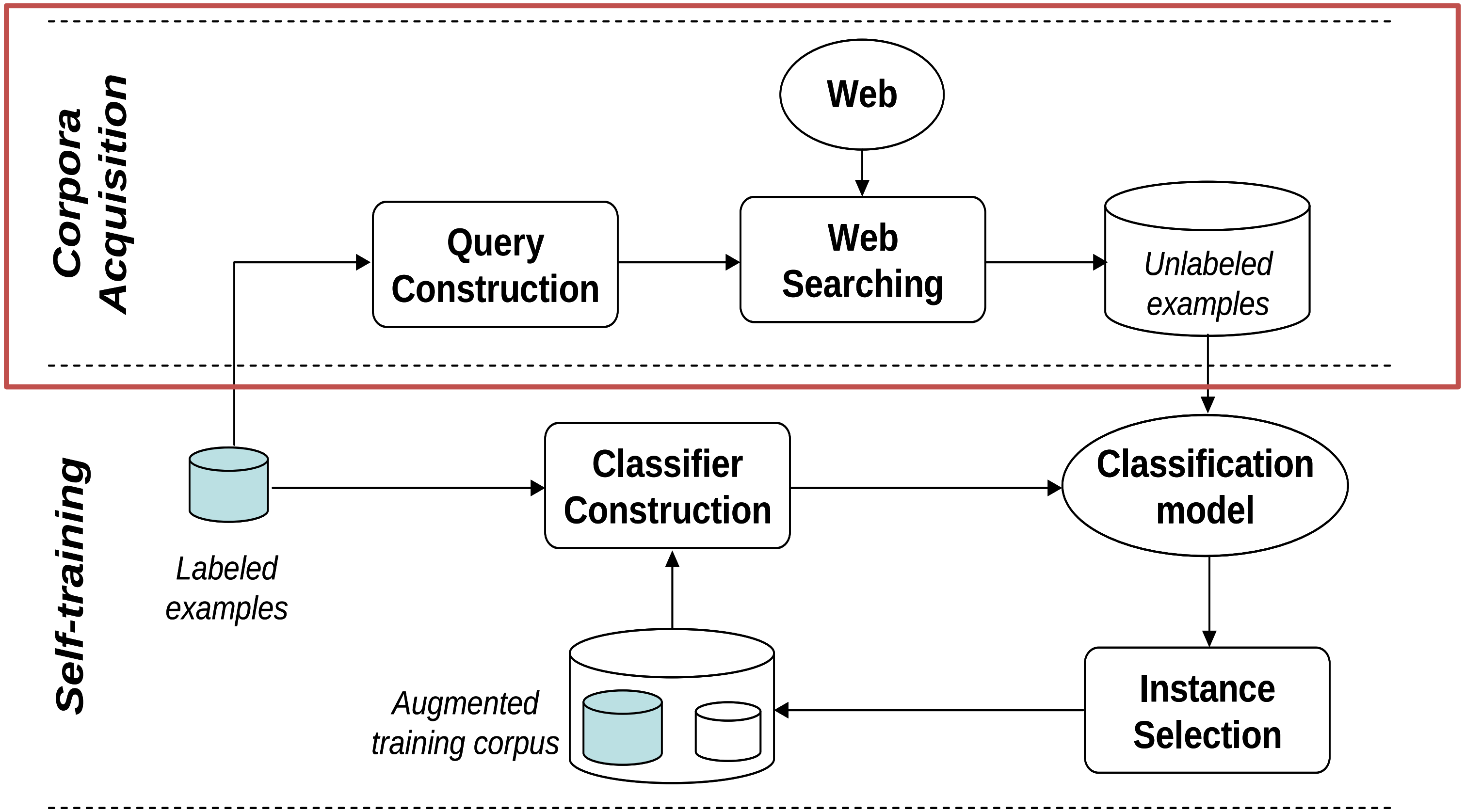
Finding unlabeled examples

- Semi-supervised methods assume the existence of a large set of unlabeled documents
  - Documents that belong to the same domain
  - Example documents for **ALL** given classes
- If unlabeled documents do not exists, then it is necessary to extract them from other place
- Idea: **using the web as corpus**, but

How to extract related documents from the Web?

How to guarantee they are relevant for the given problem?

Self-training using the Web as corpus



Rafael Guzmán-Cabrera, Manuel Montes-y-Gómez, Paolo Rosso, Luis Villaseñor-Pineda. Using the Web as Corpus for Self-training Text Categorization. *Information Retrieval*, Volume 12, Issue3, Springer 2009.

Building web queries

- Good queries are formed by **good terms**
  - Terms that helps to describe some class, and to differentiate among classes
- Good queries are **not ambiguous**
  - Long queries are very precise but have low recall; short queries tend to be ambiguous
- Proposed solution:
  - Consider frequent terms with positive IG
  - Queries of 3 terms (all possible combinations of the N best terms)

But, will be all these queries equally useful?

Collecting results from web search

Not all queries are equally relevant!

- Significance of a query $q = \{w_1, w_2, w_3\}$ to class C :

$$\Gamma_C(q) = \sum_{i=1}^{3} f_{w_i}^{C} \times IG_{w_i}$$

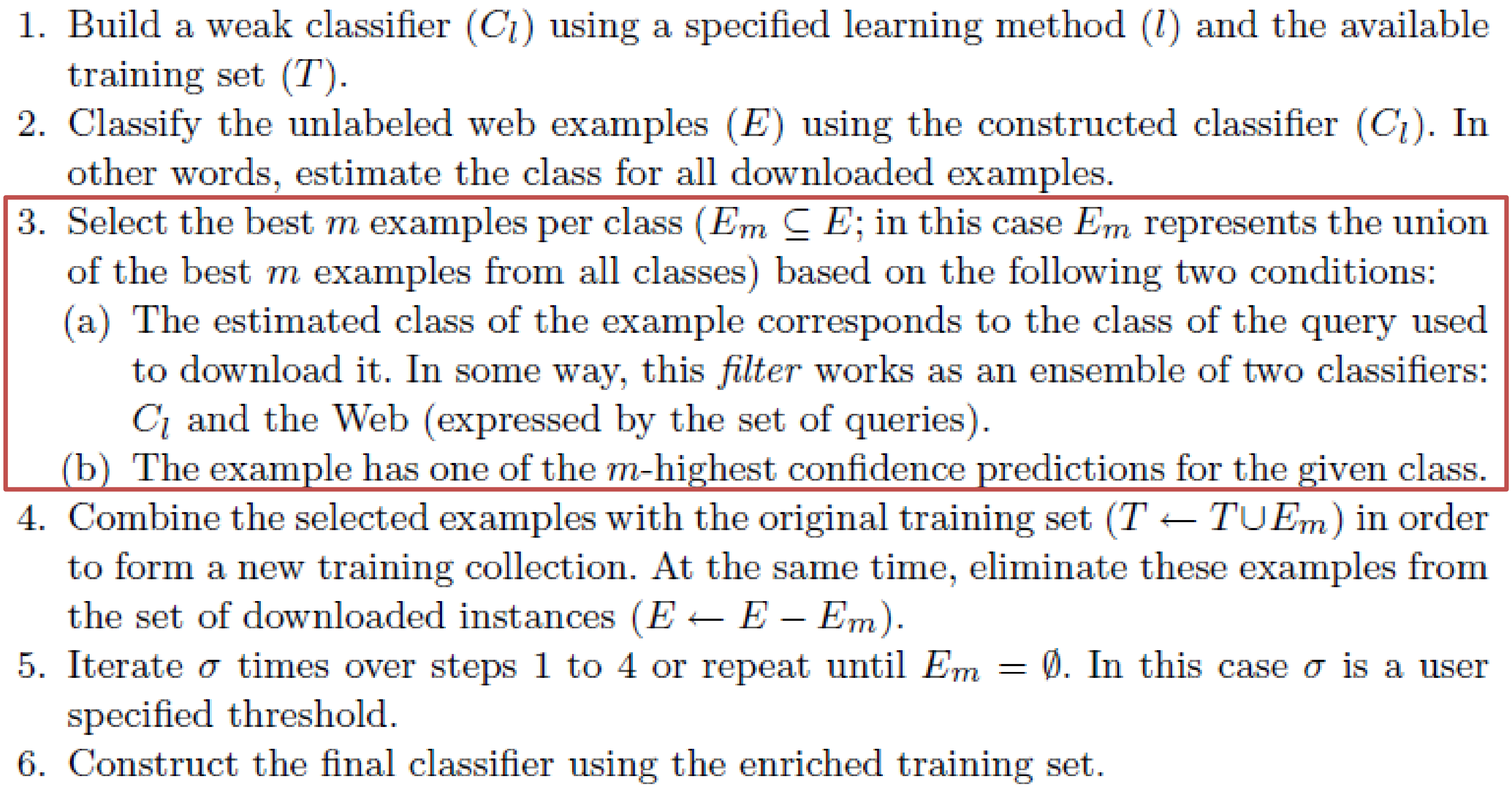
Frequency of occurrence and information gain of the query terms

- Number of downloaded examples per query in a direct proportion to its Γ-value.

$$\Psi_C(q_i) = \frac{N}{\sum_{k=1}^{M} \Gamma_C(q_k)} \times \Gamma_C(q_i)$$

Total number of snippets to be download

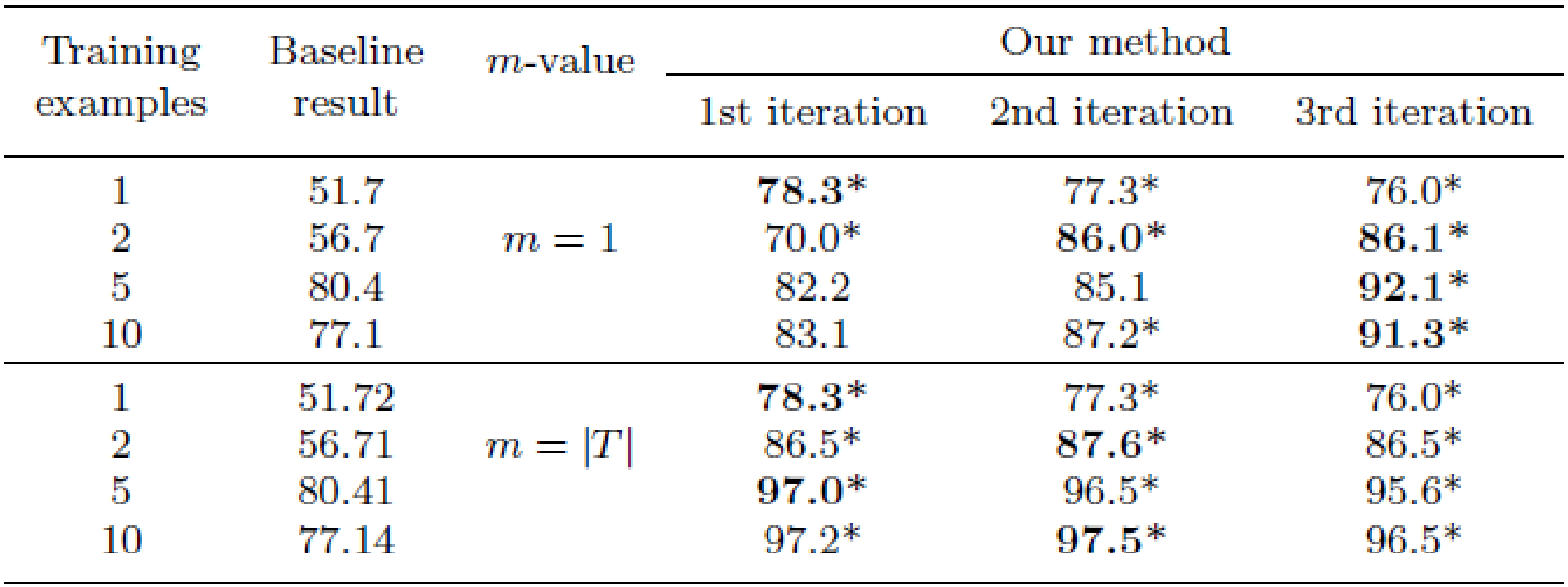
Adapted self-training procedure



1. Build a weak classifier (C_l) using a specified learning method (l) and the available training set (T).
2. Classify the unlabeled web examples (E) using the constructed classifier (C_l). In other words, estimate the class for all downloaded examples.
3. Select the best m examples per class ($E_m \subseteq E$; in this case E_m represents the union of the best m examples from all classes) based on the following two conditions:
   (a) The estimated class of the example corresponds to the class of the query used to download it. In some way, this *filter* works as an ensemble of two classifiers: C_l and the Web (expressed by the set of queries).
   (b) The example has one of the m-highest confidence predictions for the given class.
4. Combine the selected examples with the original training set ($T \leftarrow T \cup E_m$) in order to form a new training collection. At the same time, eliminate these examples from the set of downloaded instances ($E \leftarrow E - E_m$).
5. Iterate σ times over steps 1 to 4 or repeat until $E_m = \emptyset$. In this case σ is a user specified threshold.
6. Construct the final classifier using the enriched training set.

Experiment 1: Classifying Spanish news reports

Table 1 Accuracy percentages using Naïve Bayes as base classifier ($m = 1$ and $m = |T|$)

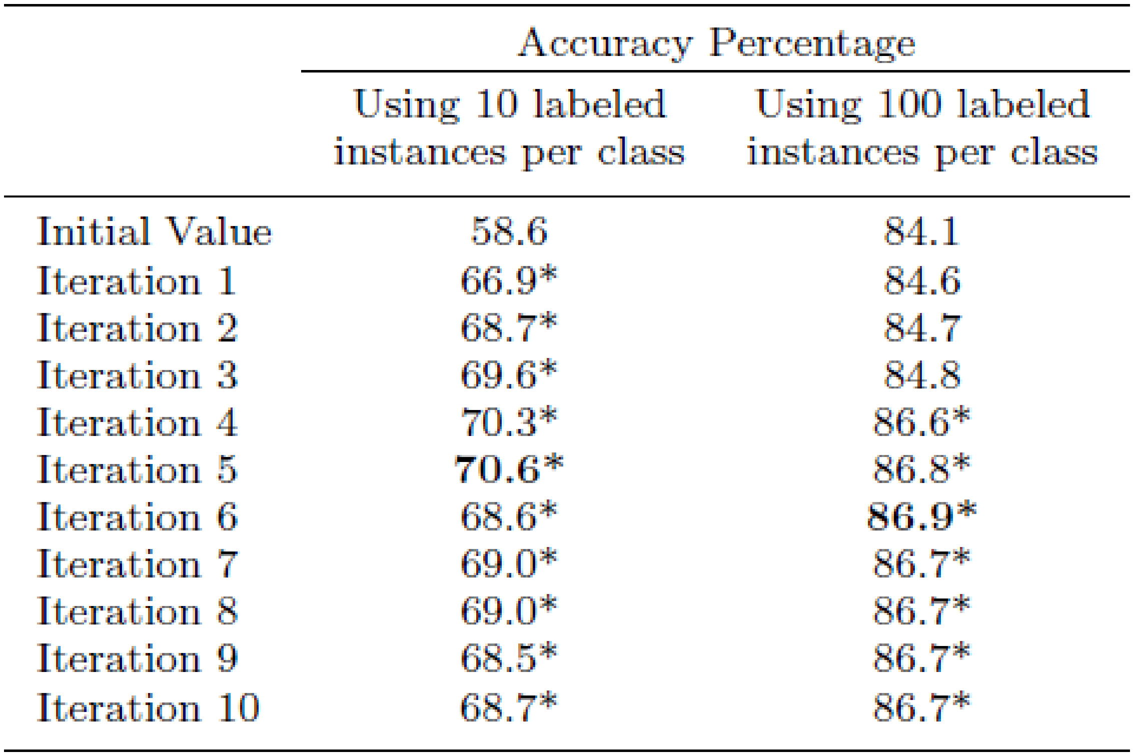


Training examples	Baseline result	m-value	Our method		
			1st iteration	2nd iteration	3rd iteration
1	51.7		**78.3***	77.3*	76.0*
2	56.7	$m = 1$	70.0*	**86.0***	**86.1***
5	80.4		82.2	85.1	**92.1***
10	77.1		83.1	87.2*	**91.3***
1	51.72		**78.3***	77.3*	76.0*
2	56.71	$m = \|T\|$	86.5*	**87.6***	86.5*
5	80.41		**97.0***	96.5*	95.6*
10	77.14		97.2*	**97.5***	96.5*

- Four classes: forest fires, hurricanes, floods, and earthquakes
- Having only **5 training instances** per class was possible to achieve a **classification accuracy of 97%**

Experiment 2: Classifying English news reports

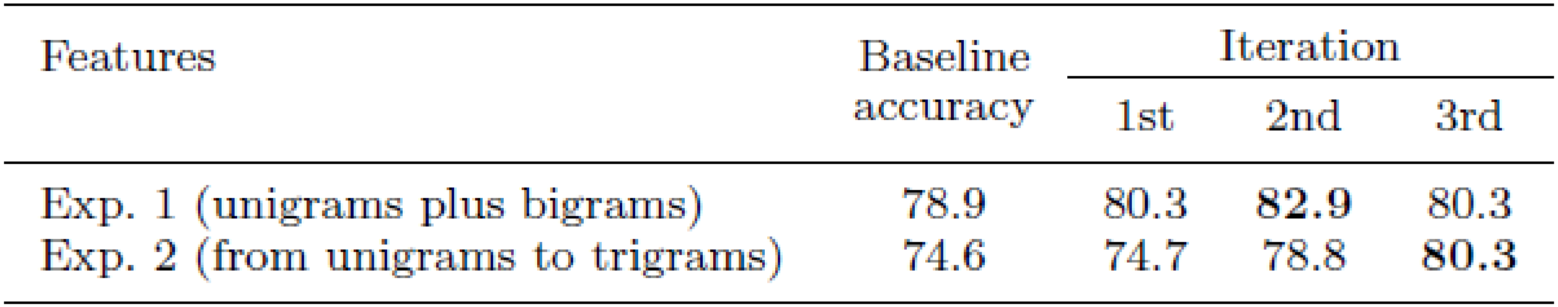
- Experiments using the R10 collection (**10 classes**); Naïve Bayes
- Higher accuracy was obtained using **only 1000 labeled examples** instead of considering the whole set of 7206 instances (84.7%)



	Accuracy Percentage	
	Using 10 labeled instances per class	Using 100 labeled instances per class
Initial Value	58.6	84.1
Iteration 1	66.9*	84.6
Iteration 2	68.7*	84.7
Iteration 3	69.6*	84.8
Iteration 4	70.3*	86.6*
Iteration 5	**70.6***	86.8*
Iteration 6	68.6*	**86.9***
Iteration 7	69.0*	86.7*
Iteration 8	69.0*	86.7*
Iteration 9	68.5*	86.7*
Iteration 10	68.7*	86.7*

Experiment 3: Authorship attribution of Spanish poems

- Poems from five different contemporary poets
  - 282 training instances, 71 test instances.
- Surprising to verify that it was feasible to extract **useful examples** from the Web **for authorship attribution**.



Features	Baseline accuracy	Iteration		
		1st	2nd	3rd
Exp. 1 (unigrams plus bigrams)	78.9	80.3	**82.9**	80.3
Exp. 2 (from unigrams to trigrams)	74.6	74.7	78.8	**80.3**

Final remarks

- Different to other semi-supervised approaches, the presented method does not require a predefined set of unlabeled examples, instead, it considers their automatic **extraction from the Web**
- Works well with very **few training examples**
  - Could be applied in classification problems having imbalanced classes, maybe in conjunction with under-sampling techniques.
- It is **domain and language independent**.
  - Experiments in three different tasks and in two different languages.

References

- Blum, A., Mitchell, T. **Combining labeled and unlabeled data with co-training**. *COLT: Proceedings of the Workshop on Computational Learning Theory*, Morgan Kaufmann, 1998, p. 92-100.
- Rafael Guzmán-Cabrera, Manuel Montes-y-Gómez, Paolo Rosso, Luis Villaseñor-Pineda. **Using the Web as Corpus for Self-training Text Categorization**. *Information Retrieval*, Volume 12, Issue3, Springer 2009.
- Rafael Guzmán-Cabrera, Manuel Montes-y-Gómez, Paolo Rosso, Luis Villaseñor-Pineda. **A Web-based Self-training Approach for Authorship Attribution**. 6th International Conference on Natural Language Processing, GoTAL 2008. Gothenburg, Sweden, August 2008.

Novel representations and methods in text classification

TEXT CLASSIFICATION USING PU-LEARNING

Outline

- One-class classification
- Taxonomy of OCC techniques
- Learning from positive and unlabeled data
- Our adaptation to PU-learning
- Experiments on opinion spam detection
- Final remarks

One class classification

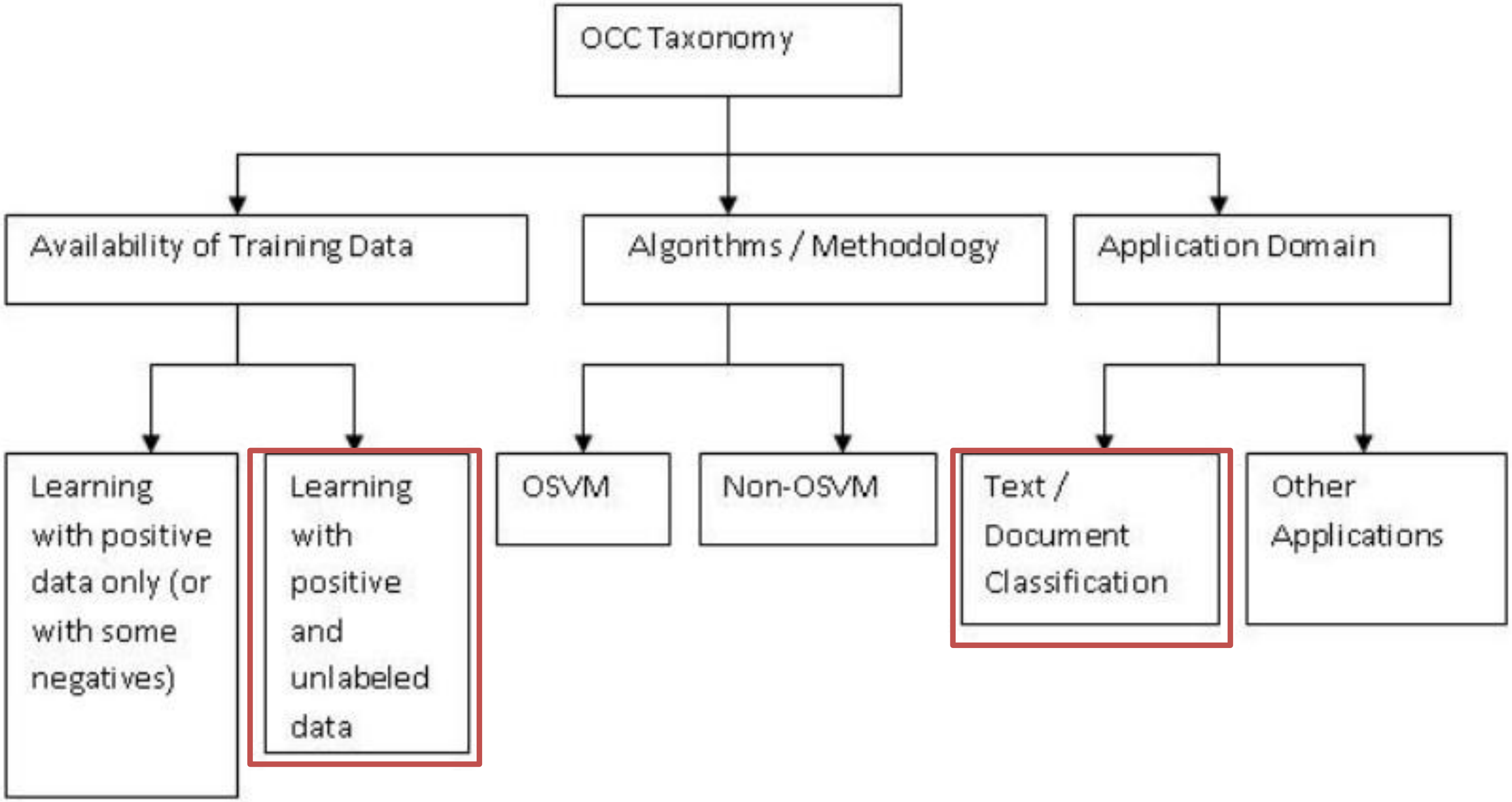
- Conventional classification algorithms classify objects into one of several pre-defined categories.
  - A problem arises when a unknown object does not belong to any of those categories.

- In OCC one of the classes is well characterized by instances in the training data; the other class, it has either **no instances at all**, very few of them, or they **do not form a representative sample** of the negative concept

An example of application

- Homepage page classification
  - Collecting sample of homepages (positive training examples) is relatively easy
  - Collecting samples of non-homepages (negative training examples) is **very challenging** because it may not represent the negative concept uniformly and may involve human bias.
- Other similar applications on textual data are:
  - Author verification
  - Wikipedia flaw detection
  - **Opinion spam detection**

Taxonomy of OCC techniques


OCC Taxonomy
Availability of Training Data
Algorithms / Methodology
Application Domain
Learning with positive data only (or with some negatives)
Learning with positive and unlabeled data
OSVM
Non-OSVM
Text / Document Classification
Other Applications

Shehroz S. Khan and Michael G. Madden. A survey of recent trends in one class classification. In *Proceedings of the 20th Irish conference on Artificial intelligence and cognitive science* (AICS'09). Dublin, Ireland, August 2009.

Learning from positive and unlabeled

- **PU-learning** is a partially supervised classification technique
  - It addresses the problem of **building a two-class classifier** with only positive and unlabeled examples.
- It is defined as a **two-step strategy**:
  - Step 1: Extract a set of negative examples called **reliable negatives** (RN) from the unlabeled examples
  - Step 2: Iteratively apply a learning algorithm on the refined training set to build a two-class classifier.

Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *Proceedings of the 18th international joint conference on Artificial intelligence* (IJCAI'03). Acapulco, Mexico, 2003.

Traditional PU-learning algorithm

- The idea is to **iteratively increase the number of unlabeled examples that are classified as negative** while maintaining the positive examples correctly classified.

1. Assign label 1 to each document in P (positive set)
2. Assign label -1 to each document in U (unlabeled set)
3. Build a classifier using P and U
4. Use the classifier to classify U
5. RN = documents in U classified as negative (reliable negatives)
6. Build a classifier using P and RN
7. Use the classifier to classify U-RN
8. Add documents classified as negative to RN
9. Repeat 6 to 8 until no more negative instances found

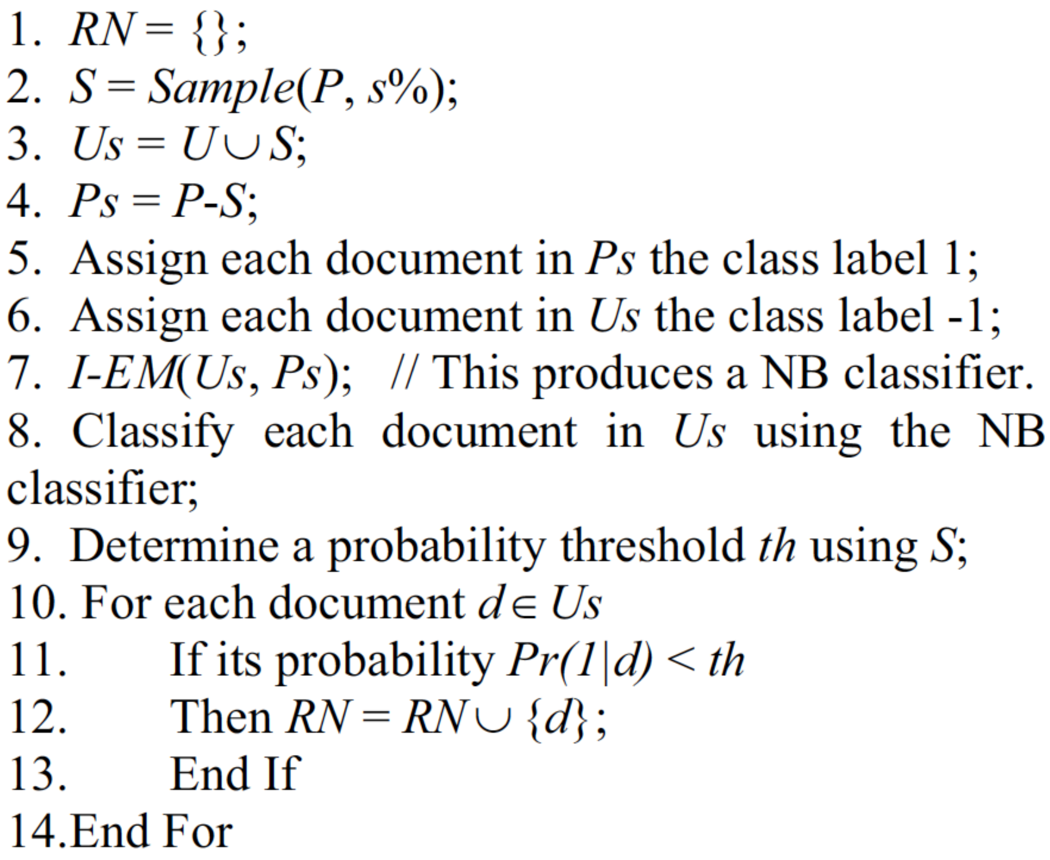
It is a self-training approach!

Alternative PU-learning approaches

- Traditional PU-learning is **very sensitive** to initial extraction of reliable negatives.
- One alternative is the **spy technique** at first step
  - Uses a subset of P as control sample, to determine a threshold to identify reliable negative instances, or to determine stop condition

Bangzuo Zhang, Wanli Zuo. Reliable Negative Extracting Based on kNN for Learning from Positive and Unlabeled Examples. *Journal of Computers*, Vol 4, No 1 (2009), 94-101, Jan 2009.

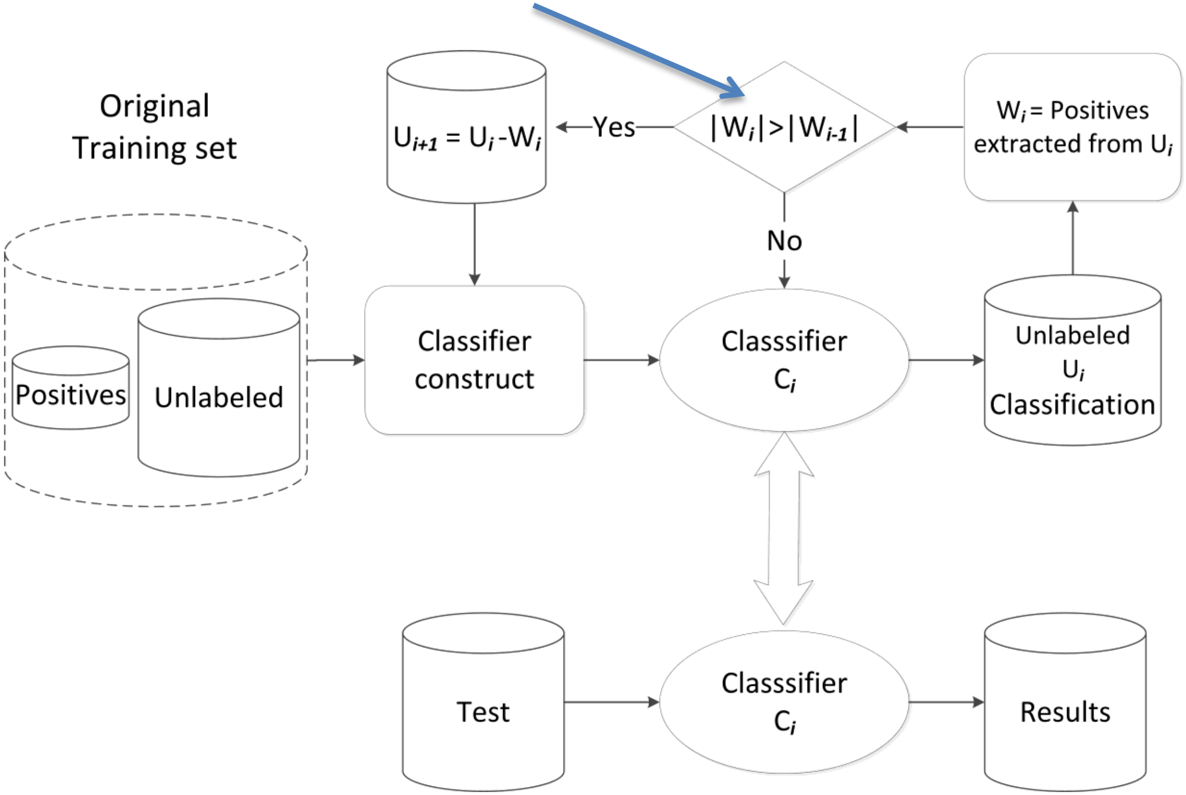
Spy technique for identifying reliable negatives



```
$RN = \{\}$;
$S = Sample(P, s\%)$;
$Us = U \cup S$;
$Ps = P\text{-}S$;
Assign each document in $Ps$ the class label 1;
Assign each document in $Us$ the class label -1;
$I\text{-}EM(Us, Ps)$;   // This produces a NB classifier.
Classify each document in $Us$ using the NB classifier;
Determine a probability threshold $th$ using $S$;
For each document $d \in Us$
    If its probability $Pr(1|d) < th$
    Then $RN = RN \cup \{d\}$;
    End If
End For
```

Our approach for PU-learning

- Instead of iteratively increase the number of reliable negative instances, **iteratively refine this set**
  - Applies a **gradual reduction** of the negative instances; at each iteration we eliminate less instances.



Using PU-Learning for opinion spam detection

- Why experiments in this domain?
  - – Large number of opinion reviews on the Web
  - – Great economic importance of online reviews
  - – **Growing trend to incorporate spam** on review sites.
    - Online reviews paid by companies to promote their products or damage the reputation of competitors
    - Ott et al. (2011) has estimated around 5% of positive hotel reviews appear to be deceptive

Ott M., Choi Y., Cardie C. and Hancock J.T. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*. Portland, Oregon, USA, 2011.

A Challenging problem

- Detecting deceptive opinions is **very difficult**
  - Opinions are typically **short texts**, written in different styles and for different purposes.
  - Human deception detection **performance is low**, with accuracies around 60% (Ott et al., 2013)

Example of a truthful opinion:

We stay at Hilton for 4 nights last march. It was a pleasant stay. We got a large room with 2 double beds and 2 bathrooms, The TV was Ok, a 27’ CRT Flat Screen. The concierge was very friendly when we need. The room was very cleaned when we arrived, we ordered some pizzas from room service and the pizza was ok also. The main Hall is beautiful. The breakfast is charged, 20 dollars, kinda expensive. The internet access (WiFi) is charged, 13 dollars/day. Pros: Low rate price, huge rooms, close to attractions at Loop, close to metro station. Cons: Expensive breakfast, Internet access charged. Tip: When leaving the building, always use the Michigan Ave exit. It’s a great view.

Example of a deceptive opinion:

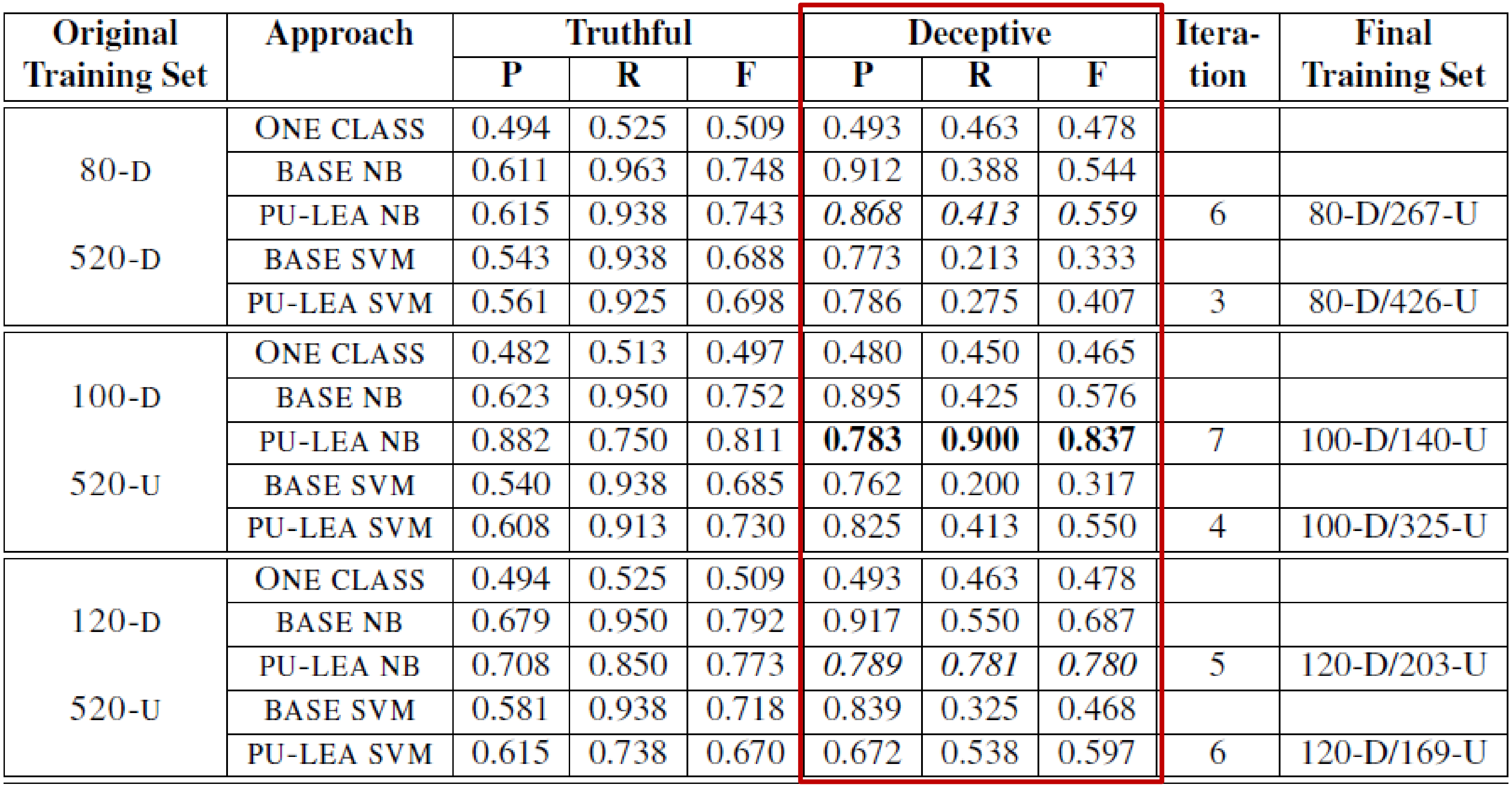
My husband and I stayed for two nights at the Hilton Chicago, and enjoyed every minute of it! The bedrooms are immaculate, and the linens are very soft. We also appreciated the free WiFi, as we could stay in touch with friends while staying in Chicago. The bathroom was quite spacious, and I loved the smell of the shampoo they provided-not like most hotel shampoos. Their service was amazing, and we absolutely loved the beautiful indoor pool. I would recommend staying here to anyone.

Experiments

- We used six different corpora
  - Test set: 80 deceptive and 80 truthful opinions.
  - Three training sets: 80, 100 and 120 positive instances, and 520 unlabeled instances (320 truthful and 200 deceptive opinions)
- Experimental setup:
  - Traditional BoW representation with binary weights
  - Naïve Bayes and SVM as base classifiers (Weka implementations; default parameters)

Donato Hernández, Rafael Gúzman, Manuel Montes, Paolo Rosso. Using PU-Learning to Detect Deceptive Opinion Spam. *4th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis* (WASSA 2013), at NAACL 2013. Atlanta, Georgia, July 2013.

Results



Original Training Set	Approach	Truthful			Deceptive			Iteration	Final Training Set
		P	R	F	P	R	F		
80-D 520-D	ONE CLASS	0.494	0.525	0.509	0.493	0.463	0.478		
	BASE NB	0.611	0.963	0.748	0.912	0.388	0.544		
	PU-LEA NB	0.615	0.938	0.743	*0.868*	*0.413*	*0.559*	6	80-D/267-U
	BASE SVM	0.543	0.938	0.688	0.773	0.213	0.333		
	PU-LEA SVM	0.561	0.925	0.698	0.786	0.275	0.407	3	80-D/426-U
100-D 520-U	ONE CLASS	0.482	0.513	0.497	0.480	0.450	0.465		
	BASE NB	0.623	0.950	0.752	0.895	0.425	0.576		
	PU-LEA NB	0.882	0.750	0.811	**0.783**	**0.900**	**0.837**	7	100-D/140-U
	BASE SVM	0.540	0.938	0.685	0.762	0.200	0.317		
	PU-LEA SVM	0.608	0.913	0.730	0.825	0.413	0.550	4	100-D/325-U
120-D 520-U	ONE CLASS	0.494	0.525	0.509	0.493	0.463	0.478		
	BASE NB	0.679	0.950	0.792	0.917	0.550	0.687		
	PU-LEA NB	0.708	0.850	0.773	*0.789*	*0.781*	*0.780*	5	120-D/203-U
	BASE SVM	0.581	0.938	0.718	0.839	0.325	0.468		
	PU-LEA SVM	0.615	0.738	0.670	0.672	0.538	0.597	6	120-D/169-U

Final remarks

- Many **real-world text classification** applications fall into the class of positive and unlabeled learning problems.
  - Negative class very generic or uncertainty on negative examples
  - Author verification, sexual predator detection
- **Good results** on the application of PU-learning to opinion spam detection (F=0.84 with 100 examples)
  - Ott et al. (2011) reported F= 0.89 using 400 positive and 400 negative instances for cross-validation.
  - Best human result in this dataset is around 60% of accuracy.

References

- Shehroz S. Khan and Michael G. Madden. **A survey of recent trends in one class classification**. In *Proceedings of the 20th Irish conference on Artificial intelligence and cognitive science* (AICS'09). Dublin, Ireland, August 2009.
- Bangzuo Zhang, Wanli Zuo. **Reliable Negative Extracting Based on kNN for Learning from Positive and Unlabeled Examples**. *Journal of Computers*, Vol 4, No 1 (2009), 94-101, Jan 2009.
- Xiaoli Li and Bing Liu. **Learning to classify texts using positive and unlabeled data**. In *Proceedings of the 18th international joint conference on Artificial intelligence* (IJCAI'03). Acapulco, Mexico, 2003.
- Donato Hernández, Rafael Gúzman, Manuel Montes, Paolo Rosso. **Using PU-Learning to Detect Deceptive Opinion Spam**. *4th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis* (WASSA 2013), at NAACL 2013. Atlanta, Georgia, July 2013.
- Ott M., Choi Y., Cardie C. and Hancock J.T. **Finding deceptive opinion spam by any stretch of the imagination**. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*. Portland, Oregon, USA, 2011.
- Myle Ott, Claire Cardie and Jeffrey T. Hancock. **Negative Deceptive Opinion Spam**. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (NAACL-HLT 2013). Atlanta, Georgia, USA, 2013-

Novel representations and methods in text classification

NEIGHBORHOOD CONSENSUS FOR TEXT CATEGORIZATION

Outline

- Collective classification
  - Motivation and definition
  - Approaches for hypertext classification
- Text classification using neighborhood information
  - Experiments on short text classification
  - Experiments on crosslingual classification
- Final remaks

Collective classification (motivation)

- Traditional text classification methods:
  - Represent **each** document by a feature (word) vector
  - Learn a classifier based on manually labeled training data
  - Apply the classifier to each unlabeled document in a "**context-free**" manner.
- Decisions are based only on the information contained in the given test document, **disregarding the other documents** in the test set.

Angelova, R., & Weikum, G. Graph-based text classification: Learn from your neighbors. In Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '06. Seattle, WA, USA, 2006.

Collective classification (general idea)

- Not only determine the topic of a single document, but to infer it for a **collection of documents**.
  - This is the real application scenario for a text classifier
- Try to collectively optimise this problem taking into account the **connections present** among the documents, for example:
  - Papers citing papers
  - Links among web pages (**hypertext**)
  - Other relations such as: same author, same conference, **similar content**, etc.

Approaches for hypertext classification (1)

- **Straightforward approach**: Incorporate words of the neighbors into the vector of the given document
  - Adjust the non-zero weights of existing terms in the original vector
  - Bring in new terms from the neighbors (i.e, expand the document)
- Generally it does not lead to a robust solution.
  - Parameter tuning is problematic

Hyo-Jung Oh, Sung Hyon Myaeng, and Mann-Ho Lee. A practical hypertext catergorization method using links and incrementally available class information. In Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval (SIGIR '00). Athens, Greece, 2000.

Approaches for hypertext classification (2)

- **Local approaches**: learn a model locally, without considering unlabeled data, and then apply the model iteratively to classify unlabeled data.
  - At each iteration, the label of each document is influenced by the popularity of this label among their neighbors
- **Global approaches**: aim to estimate the labels of all test documents simultaneously, by modeling the mutual influence between neighboring documents.
  - Based on global optimization techniques
  - Tend to exploit the **links occurring between labeled and unlabeled** data for learning

Neighborhood consensus classification (NCC)

- Supported on the idea that similar documents may belong to the same category.
  - Classifies documents by considering their **own information** as well as information about the **category assigned to other similar documents** from the same target collection
- Does not need information about the association between documents and can be easily **combined with different classification algorithms**.

Gabriela Ramírez-de-la-Rosa, Manuel Montes-y-Gómez, Thamar Solorio, Luis Villaseñor-Pineda. A document is known by the company it keeps: Neighborhood consensus for short text categorization. *Journal of Language Resources and Evaluation*. Vol. 47, Issue 1, March 2013.

A reclassification approach

- It is a local but **not iterative** approach
  - Learns a model locally, and **classifies each document individually**

$$\text{class}(d) = \arg\max_{c_j \in \mathbb{C}} (\gamma(d, c_j))$$

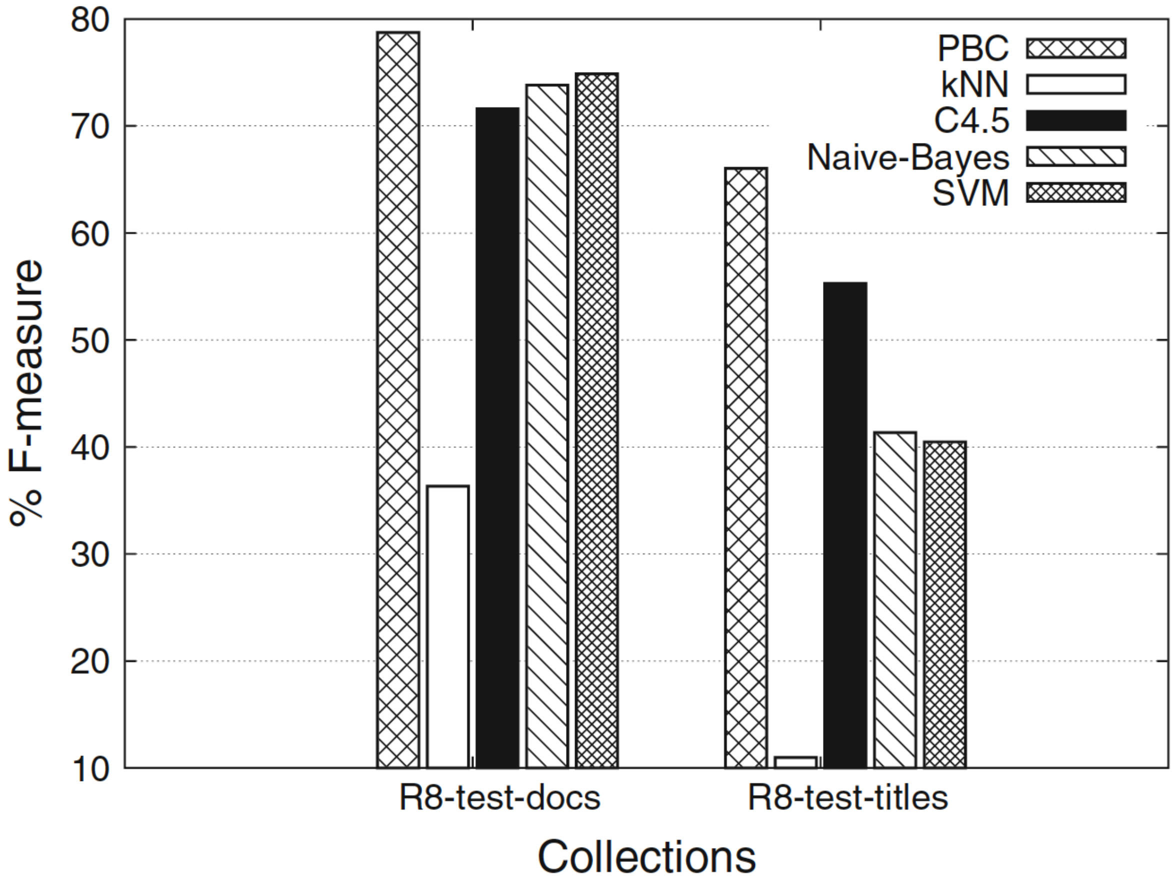
  - Finds the N more similar documents in the target set
    - Content similarity (cosine function); KNN
  - **Re-labels the documents** considering the categories of their neighbors (similar documents)

$$\text{class}(d) = \arg\max_{j} \left(\gamma(d, c_j) + \frac{1}{|\mathbb{N}_k^d|} \sum_{d_i \in \mathbb{N}_k^d} \gamma(d_i, c_j) \right)$$

Experiments

- **Short documents** are difficult to categorize since they contain a small number of words whose absolute frequency is relatively low
  - Produce very sparse representations
- The goal is to evaluate the effectiveness of NCC in the classification of short documents
  - Classification of complete news articles
  - Classification of news titles (short texts)

Complexity of short text classification



- **Prototype-based classification** emerged as the most robust classification approach for short documents

NCC using prototype-based classification

$$\mathrm{class}(d) = \arg\max_{j} \left(\gamma(d, c_j) + \frac{1}{|\mathbb{N}_k^d|} \sum_{d_i \in \mathbb{N}_k^d} \gamma(d_i, c_j) \right)$$

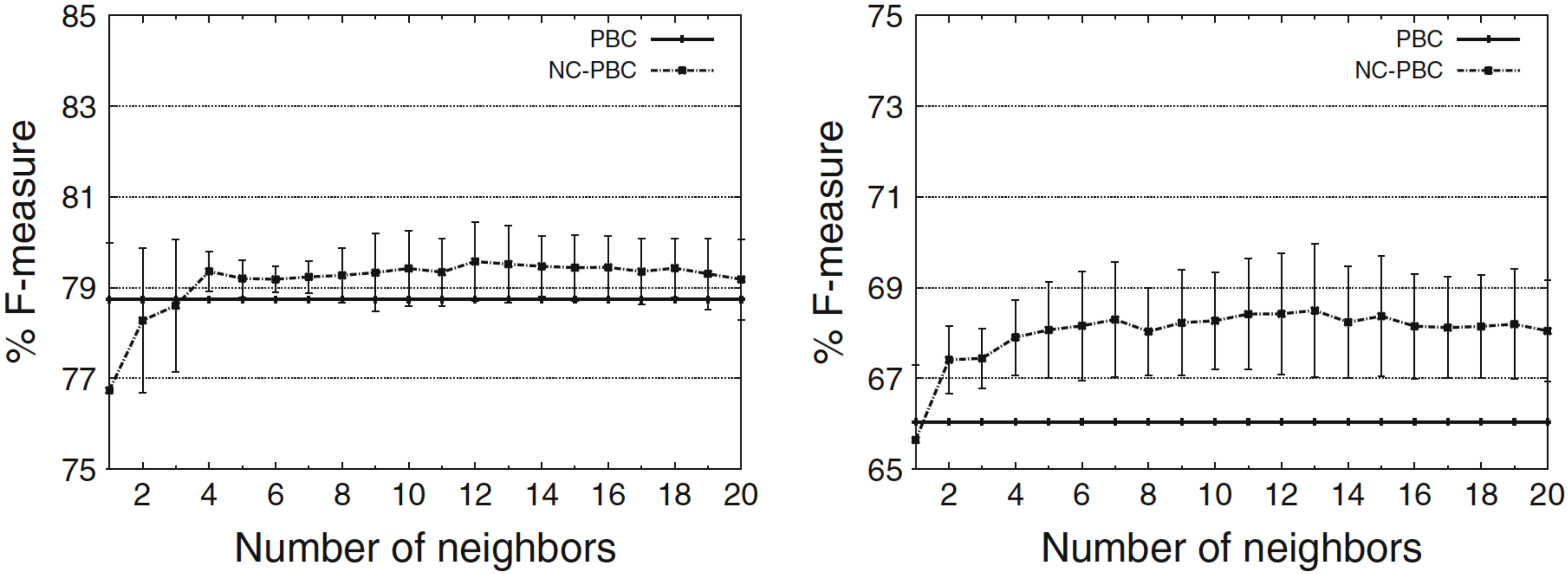
$$\mathrm{class}(d) = \arg\max_{j} \left(\lambda \mathrm{sim}(d, P_j) + (1 - \lambda) \frac{1}{|\mathbb{N}_k^d|} \sum_{d_i \in \mathbb{N}_k^d} \left[\mathrm{influence}(d_i, d) \times \mathrm{sim}(d_i, P_j) \right] \right)$$

- Prototypes are the **centroids** of the categories

$$P_i = \frac{1}{|| \sum_{d \in c_i} d ||} \sum_{d \in c_i} d$$

- Similarity among documents and between prototypes and documents is computed using the **cosine** formula
- *K* = number of used neighbors; *lambda* = relative importance of neighbors information

Short-text classification using NC-PBC

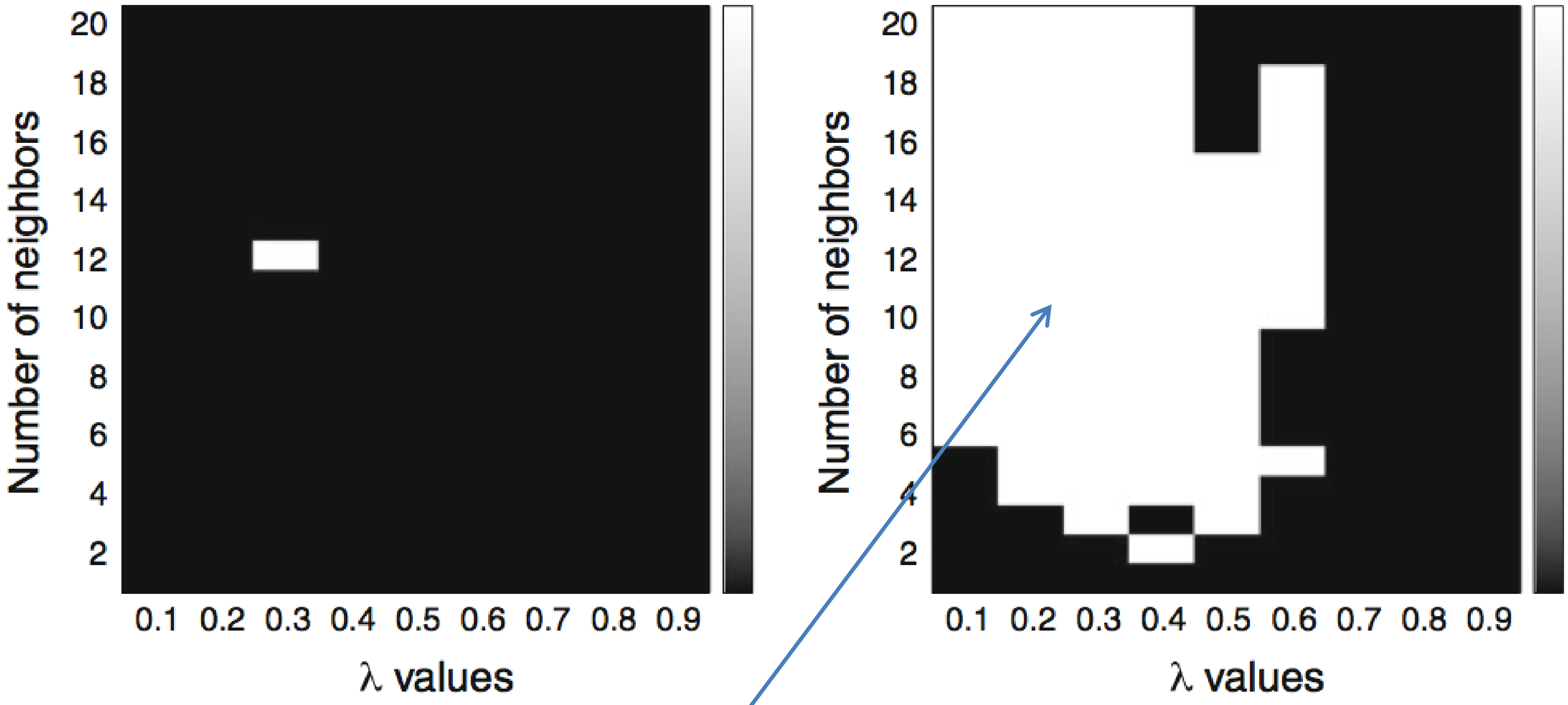


(a) R8-test-docs

(b) R8-test-titles

- Information from the neighbors **improved the classification** performance of short texts.
- It was not very useful in the case of regular-length documents

Parameter tuning



(a) R8-test-docs **(b)** R8-test-titles

- For all these combinations of parameter values, results of NC-PBC **significantly outperformed** the results of PBC
- Use more than 5 neighbors and give them too much importance in the final decision

Additional experiments

- A major difficulty of supervised techniques is that they commonly **require large training sets**
- For many applications in several languages these datasets are extremely small or, what is worst, they are **not available**
- One solution: **crosslingual text classification**
  - Consists in exploiting labeled documents in a source language to classify documents in a different target language

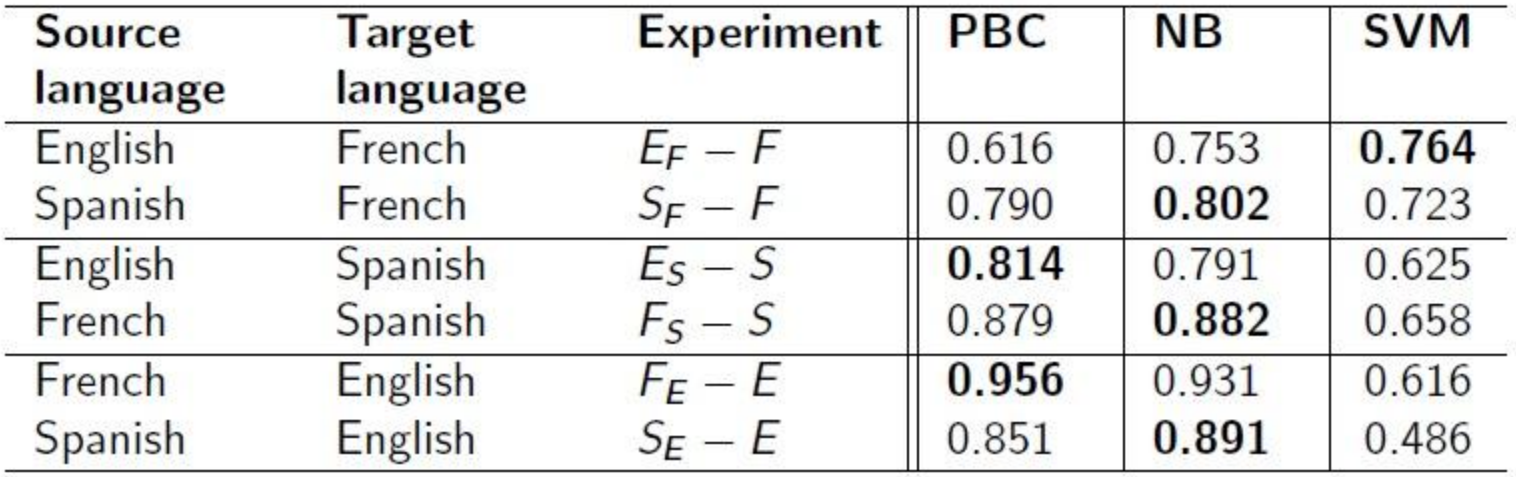
NCC in crosslingual text classification

- Common approach: **Translate training documents** to target language using translation machines.
  - The resulting classifier is a **weak classifier**, because of translation errors as well as cultural discrepancies manifested in both languages.
- The purpose of the experiment is to evaluate the improvement in the classification performance of these weak classifiers by **applying the neighborhood consensus classification** approach.

Gabriela Ramírez, Manuel Montes, Luis Villaseñor, David Pinto, Thamar Solorio. Using Information from the Target Language to Improve Crosslingual Text Classification. 7th International Conference on Natural Language Processing IceTAL-2010, Reykjavik, Iceland, August 2010.

Baseline results

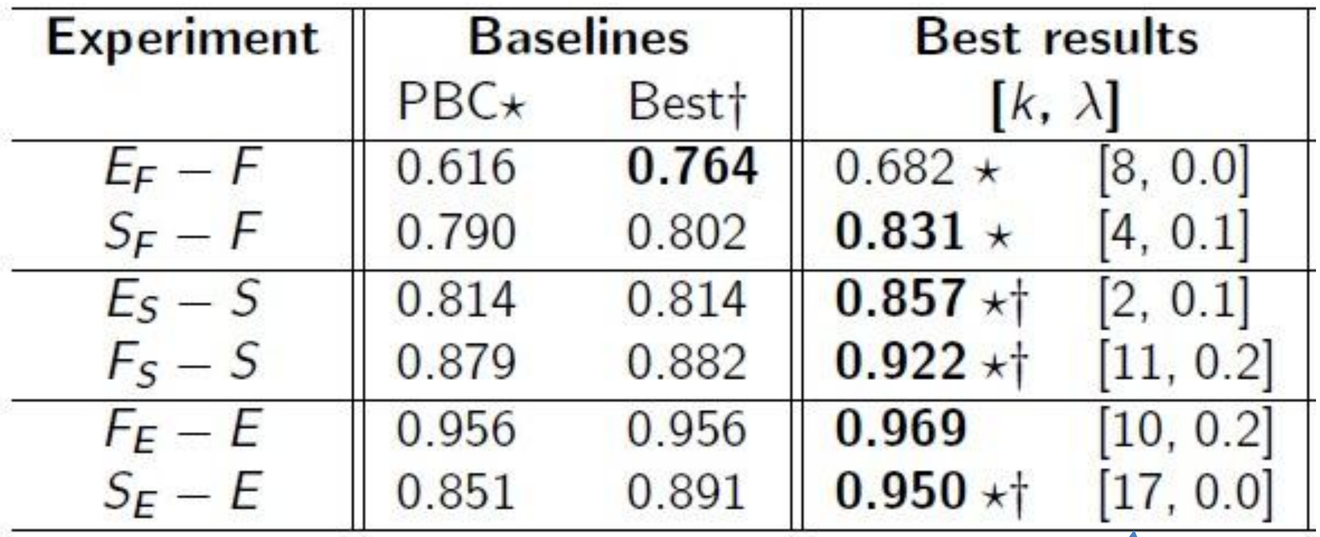
- **Three languages** (English, French and Spanish)
- News reports corresponding to **four classes**: crime, disasters, politics, and sports.
- For each language we used 320 documents; 80 per each class



Source language	Target language	Experiment	PBC	NB	SVM
English	French	$E_F - F$	0.616	0.753	**0.764**
Spanish	French	$S_F - F$	0.790	**0.802**	0.723
English	Spanish	$E_S - S$	**0.814**	0.791	0.625
French	Spanish	$F_S - S$	0.879	**0.882**	0.658
French	English	$F_E - E$	**0.956**	0.931	0.616
Spanish	English	$S_E - E$	0.851	**0.891**	0.486

Results of NCC

- Demonstrate the usefulness of considering information from target language (dataset) to reclassify the documents



Experiment	Baselines		Best results	
	PBC$\star$	Best$\dagger$		$[k, \lambda]$
$E_F - F$	0.616	**0.764**	0.682 $\star$	[8, 0.0]
$S_F - F$	0.790	0.802	**0.831** $\star$	[4, 0.1]
$E_S - S$	0.814	0.814	**0.857** $\star\dagger$	[2, 0.1]
$F_S - S$	0.879	0.882	**0.922** $\star\dagger$	[11, 0.2]
$F_E - E$	0.956	0.956	**0.969**	[10, 0.2]
$S_E - E$	0.851	0.891	**0.950** $\star\dagger$	[17, 0.0]

- **More neighbors** than in short text classification
- Also, **greater importance to the neighbors**

Final remarks

- NCC determines the category of documents by taking advantage of the information about the relationships between documents from the **same target collection**
- **Effective to improve the classification** performance in complex scenarios:
  - Short text classification
  - Crosslingual text classification
  - Learning from small training sets
- **Performance is robust** for different parameter values, but better results were obtained when using more than ten neighbors and small lambda values.

References

- Angelova, R., & Weikum, G. **Graph-based text classification: Learn from your neighbors**. *In Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, SIGIR ’06. Seattle, WA, USA, 2006.
- Carlos Laorden, Borja Sanz, Igor Santos, Patxi Galán-García, and Pablo G. Bringas. **Collective classification for spam filtering**. *In Proceedings of the 4th international conference on Computational intelligence in security for information systems* (CISIS'11), Málaga Spain, 2011.
- Qing Lu and Lise Getoor. **Link-based Text Classification**. *IJCAI Workshop on Text Mining and Link Analysis*. Acapulco, Mexico, 2003.
- Hyo-Jung Oh, Sung Hyon Myaeng, and Mann-Ho Lee. **A practical hypertext catergorization method using links and incrementally available class information**. *In Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval* (SIGIR '00). Athens, Greece, 2000.
- Gabriela Ramírez-de-la-Rosa, Manuel Montes-y-Gómez, Thamar Solorio, Luis Villaseñor. **A document is known by the company it keeps: Neighborhood consensus for short text categorization**. *Journal of Language Resources and Evaluation*. Vol. 47, Issue 1, March 2013.
- Gabriela Ramírez, Manuel Montes, Luis Villaseñor, David Pinto, Thamar Solorio. **Using Information from the Target Language to Improve Crosslingual Text Classification**. *7th International Conference on Natural Language Processing* IceTAL-2010, Reykjavik, Iceland, August 2010.