Special Topics in Text Mining

Manuel Montes y Gómez

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

mmontesg@inaoep.mx

Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico.

Non-conventional methods for text classification

Introduction

- Text classification is the assignment of free-text documents to one or more predefined categories based on their content
- Important to remember:
  - Assigns documents to known categories
    - It does not aim to discover topics or classes
  - It is a supervised task: training data is required

Topics of this session

Text classification research

Improve document representation	Adapt/design classification methods	Search for New applications

- *Commonly: supervised learning*
- Methods specially suited to difficult scenarios
  - Few training examples
  - Lack of negative training examples
  - Lack of examples in target language
  - Lack of examples in target domain

Topics of this session

- Semi-supervised learning
  - Self-training and co-training
  - Using the Web as corpus
- One-class classification
  - Learning from positive and unlabeled data
- Set-based classification
  - Neighborhood classification
- Multilingual text classification

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

How to deal with these problems?

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

Initially it trains the classifier with available data: " a weak classifier"

Procedure Selftraining (L_O, U)

1 L_O is labeled data; U is unlabeled data
2 $c \leftarrow$ train(L_O)
3 **Loop until** stopping criteria is met
4 $L \leftarrow L_O$ + select(Label(U, c)
5 $c \leftarrow$ train(L)
6 **End loop**
7 **Return** c

Classify unlabeled data using the weak classifier

Selects best instances to be incorporated into the training set

How to select the most confident instances?

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)

1 L is labeled data, U is unlabeled data
2 $P \leftarrow$ random selection from U
3 **Loop until** stopping criteria is met
4 $F_1 \leftarrow$ train(view$_1$(L))
5 $F_2 \leftarrow$ train(view$_2$(L))
6 $L \leftarrow L$ + select(label(P, F_1) + select(labelP, F_2))
7 Remove the labeled instances from P and replenish P from U
8 **end loop**

Trains two classifiers using the same data but different features

Selects best instances, from both classifiers to be incorporated into the training set

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

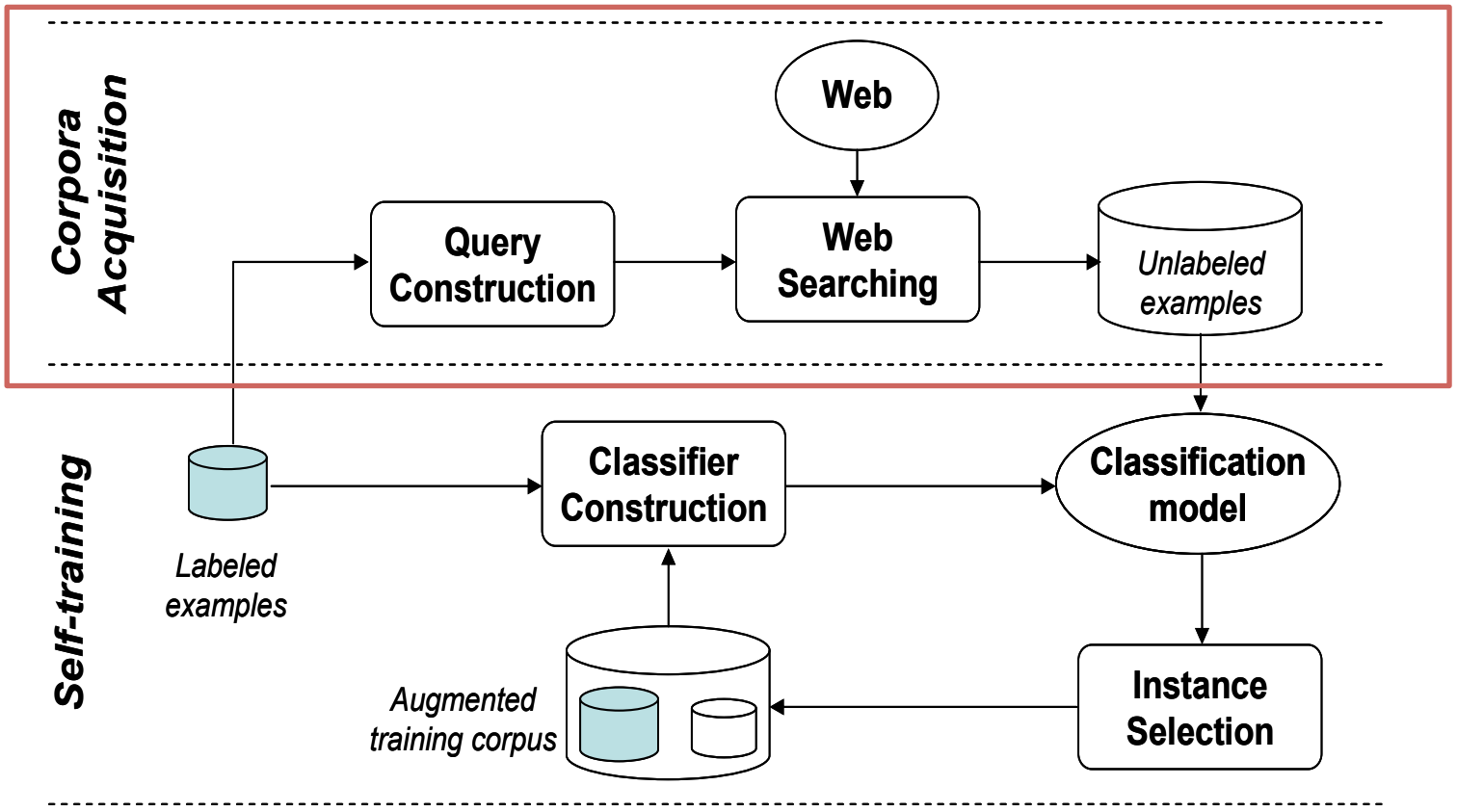
Finding unlabeled examples

- 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 are not available, 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

Relevant words of wheat class (R8 collection)

Tabla 4.1: Palabras relevantes para la categoría *wheat*

Palabra	f	IG	f*IG
wheat	418	199.34	83325.87
grain	164	116.65	19131.12
tonnes	210	55.56	11668.86
corn	89	39.51	3516.92
agriculture	75	18.52	1389.37
trade	42	12.17	511.20
export	76	10.67	810.95
usda	54	7.64	412.61
crop	51	5.97	304.57
washington	44	5.79	255.16

Used queries for wheat class (R8 collection)

Tabla 4.4: Número de ejemplos no etiquetados descargados por petición.

Petición	w_1	w_2	w_3	Γ_S	Ψ_S
wheat grain tonnes	83,325.88	19,131.12	11,668.86	114,125.86	133
wheat grain agriculture	83,325.88	19,131.12	1,389.38	103,846.38	121
wheat grain export	83,325.88	19,131.12	810.95	103,267.95	120
wheat grain crop	83,325.88	19,131.12	304.58	102,761.58	120
wheat tonnes corn	83,325.88	11,668.86	304.58	95,299.31	111
wheat tonnes trade	83,325.88	11,668.86	511.21	95,505.94	111
wheat tonnes usda	83,325.88	11,668.86	412.61	95,407.35	111
wheat tonnes washington	83,325.88	11,668.86	255.16	95,249.9	111
grain tonnes export	19,131.12	11,668.86	810.95	31,610.94	37
grain agriculture trade	19,131.12	1,389.38	511.21	21,031.71	25
				858,106.92	1,000

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	$m = 1$	**78.3***	77.3*	76.0*
2	56.7		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	$m = \|T\|$	**78.3***	77.3*	76.0*
2	56.71		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*

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.

One class classification

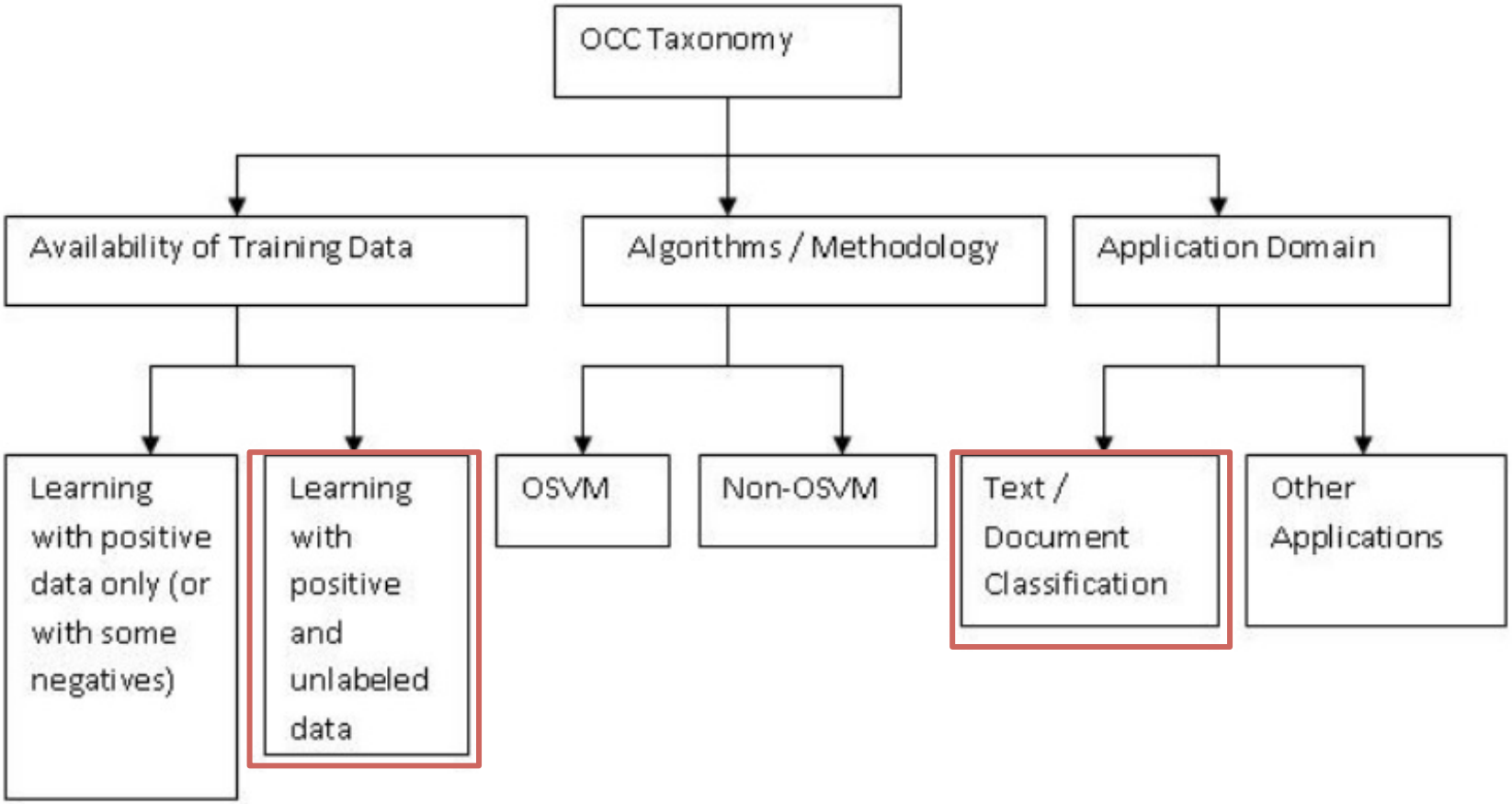
- Conventional classification algorithms classify objects into one of several pre-defined categories.
  - A problem arises when a unknown object does not belong to any of those categories.
- In OCC one of the classes is well characterized by instances in the training data; the other class, it has either **no instances at all**, very few of them, or they **do not form a representative sample** of the negative concept.

How to deal with this situation?

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
  - Opinion spam detection

Taxonomy of OCC techniques



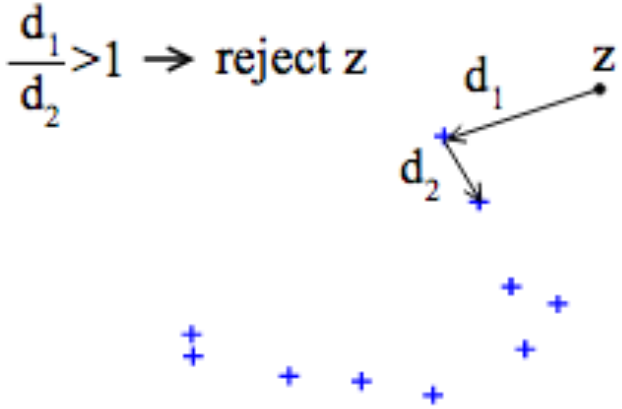
Shehroz S. Khan and Michael G. Madden. A survey of recent trends in one class classification. In *Proceedings of the 20th Irish conference on Artificial intelligence and cognitive science* (AICS'09). Dublin, Ireland, August 2009.

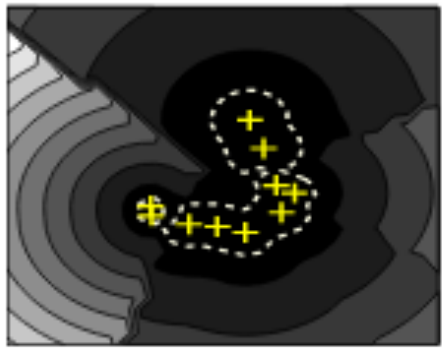
One class KNN

- Called Nearest Neighbor Description (NN-d)
- A test object z is accepted as a member of target class provided that its local density is greater than or equal to the local density of its nearest neighbor in the training set.
- Different numbers of nearest neighbors can be considered.
  - More neighbors will make the method less sensitive to noise.

The one-class 1-NN

$$f_{NN^{tr}}(z) = I\left(\frac{\|z - NN^{tr}(z)\|}{\|NN^{tr}(z) - NN^{tr}(NN^{tr}(z))\|}\right)$$





SHEHROZ S.KHAN, MICHAEL G.MADDEN. One-Class Classification: Taxonomy of Study and Review of Techniques. The Knowledge Engineering Review, pp 1-30, 2014.

One class Naïve Bayes

- The probability of d to belong to c is calculated as:

$$P(c|d) = P(c)\prod_{i=1}^{M} P(t_i|c) \qquad P(t_i|c) = \frac{1 + N_i}{M + \sum_{k=1}^{|c|} N_k}$$

- Document d is assigned to c if its probability to belong to c is greater than the minimum probability from the training set:

$$P(c|d) > \delta \qquad \delta = \min\left(\forall d \in c : P(c|d)\right)$$

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

1. $RN = \{\}$;
2. $S = Sample(P, s\%)$;
3. $Us = U \cup S$;
4. $Ps = P\text{-}S$;
5. Assign each document in Ps the class label 1;
6. Assign each document in Us the class label -1;
7. $I\text{-}EM(Us, Ps)$; // This produces a NB classifier.
8. Classify each document in Us using the NB classifier;
9. Determine a probability threshold th using S;
10. For each document $d \in Us$
11. If its probability $Pr(1|d) < th$
12. Then $RN = RN \cup \{d\}$;
13. End If
14. End For

A sample of P is inserted in U, and used as a control set.

The control set is used to determine a threshold to select the reliable negative intances
Non instance of S has to be included in RN

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 three 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
  - SVM as base classifier (Weka; default parameters)

Donato Hernández, Rafael Gúzman, Manuel Montes, Paolo Rosso. Detecting positive and negative deceptive opinions using PU-learning. Information Processing and Management 51 (2015) 433–443

Results

Table 1
Detailed results on the classification of *positive* opinions using 60, 80, 100 and 120 labeled deceptive opinions (DP) and 520 of unlabeled examples (UN) for training. In this table, P, R and F state for precision, recall and f-measure respectively; results in bold indicate the best performance by the proposed method.

Initial training set	Used method	Deceptive			Truthful			General f-measure	# of iterations	Final training set
		P	R	F	P	R	F			
60-DP/520-UN	BASELINE	0.896	0.268	0.408	0.605	0.975	0.746	0.577	1	60-DP/520-UN
	PU-L ORIGINAL	0.878	0.275	0.413	0.572	0.965	0.718	0.566	2	60-DP/473-UN
	PU-L MODIFIED	0.895	0.298	0.441	0.581	0.968	0.726	0.584	4	60-DP/394-UN
80-DP/520-UN	BASELINE	0.921	0.330	0.482	0.593	0.973	0.736	0.609	1	80-DP/520-UN
	PU-L ORIGINAL	0.925	0.363	0.519	0.604	0.970	0.744	0.632	2	80-DP/450-UN
	PU-L MODIFIED	0.842	0.415	0.547	0.618	0.933	0.742	0.645	7	80-DP/253-UN
100-DP/520-UN	BASELINE	0.919	0.408	0.561	0.621	0.965	0.756	0.689	1	100-DP/520-UN
	PU-L ORIGINAL	0.926	0.420	0.575	0.627	0.968	0.760	0.668	2	100-DP/432-UN
	PU-L MODIFIED	**0.852**	**0.728**	**0.780**	**0.768**	**0.868**	**0.811**	**0.796**	8	100-DP/112-UN
120-DP/520-UN	BASELINE	0.931	0.453	0.606	0.640	0.968	0.770	0.705	1	120-DP/520-UN
	PU-L ORIGINAL	0.916	0.480	0.626	0.648	0.955	0.772	0.699	2	120-DP/425-UN
	PU-L MODIFIED	0.803	0.700	0.743	0.738	0.823	0.774	0.759	7	120-DP/144-UN

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-

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 classification**)
  - 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

- 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 (test) 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**

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

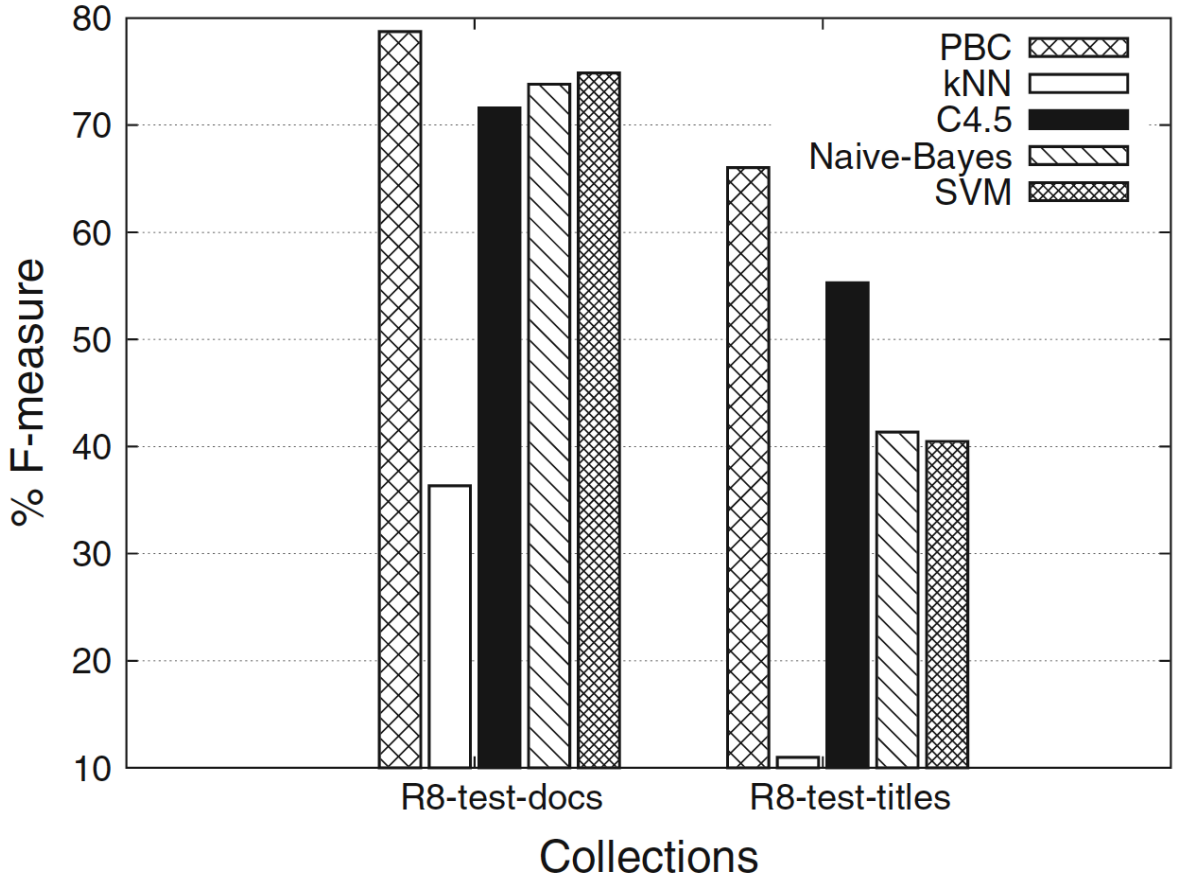
  - Finds the N more similar documents in the target set
    - Content similarity (cosine function); KNN
  - **Re-labels the documents** considering the categories of their neighbors (similar documents)

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

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

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

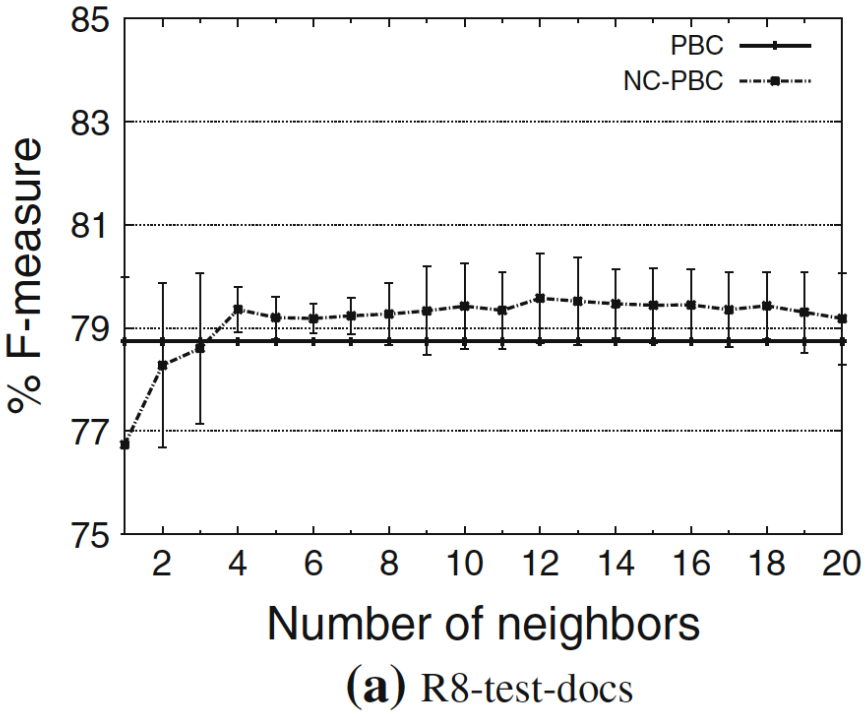
$$\text{class}(d) = \arg\max_j \left(\lambda \text{sim}(d, P_j) + (1 - \lambda) \frac{1}{|\mathbb{N}_k^d|} \sum_{d_i \in \mathbb{N}_k^d} \left[\text{influence}(d_i, d) \times \text{sim}(d_i, P_j) \right] \right)$$

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

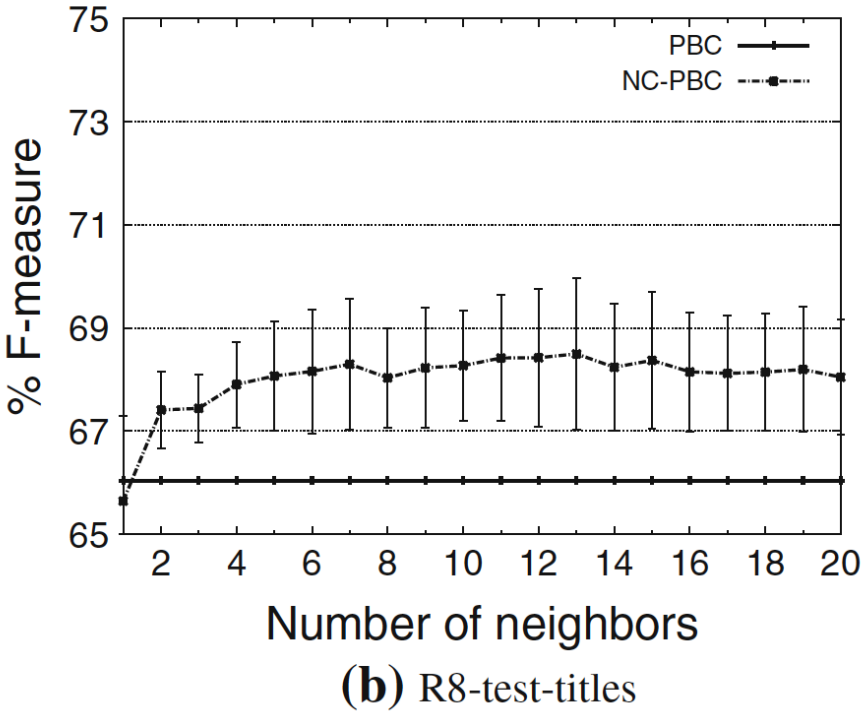
$$P_i = \frac{1}{\| \sum_{d \in c_i} d \|} \sum_{d \in c_i} d$$

- 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

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
  - 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.

Multilingual text classification

Agenda

- Multilinguism data/problem
- Poly-lingual text classification
  - Language identification
- Cross-lingual text classification
  - Using machine translation
  - Employing multilingual dictionaries or ontologies
- Re-categorization methods

Initial questions

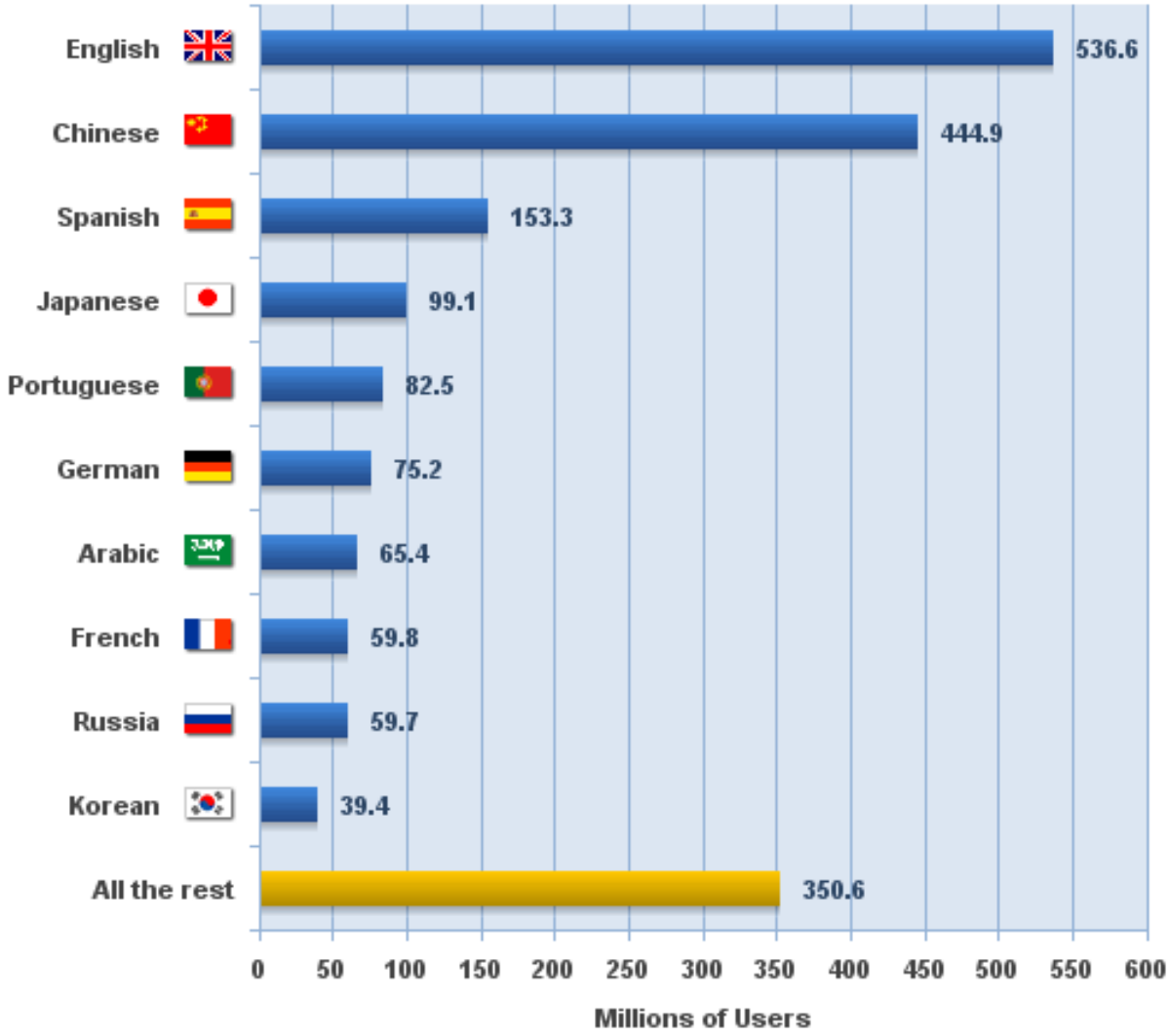
- What is multilingual text classification?
- Is it concerns a practical problem?
- How to build a multilingual text classification system?
- Which multilingual resources are necessary?
- Equally difficult for all language combinations?

Languages in the world

- It is difficult to give an exact figure of the number of languages that exist in the world
  - Not always easy to differentiate between language and dialect.
- It is usually estimated that the number of languages in the world varies between 3,000 and 8,000.

Pos	Language	Family	Script(s) Used	Speakers (Millions)
1	**Mandarin**	Sino-Tibetan	Chinese Characters	1151
2	**English**	Indo-European	Latin	1000
3	**Spanish**	Indo-European	Latin	500
4	**Hindi**	Indo-European	Devanagari	490
5	**Russian**	Indo-European	Cyrillic	277
6	**Arabic**	Afro-Asiatic	Arabic	255
7	**Portuguese**	Indo-European	Latin	240
8	**Bengali**	Indo-European	Bengali	215
9	**French**	Indo-European	Latin	200
10	**Malay, Indonesian**	Malayo-Polynesian	Latin	175

Languages in the Web (users)



Importance of handling multilingual data

- Existence of a multilingual worldwide network
  - Representation of English is now less than 40%
- The time of globalization is coming; many countries have been unified.
  - Example: European Union
- In addition, many countries adopt multiple languages as their official languages
  - Example: Moroco
- New technologies in network infrastructure and Internet set the platform of the cooperation and globalization.

Multilingual text classification

- Poly-lingual classification
  - The system is trained using labeled documents from all the different languages, and allows to classify documents from any of these languages.
- Cross-lingual classification
  - The system use labeled training data for only one language to classify documents in other languages.

Ideas for achieving these two approaches?

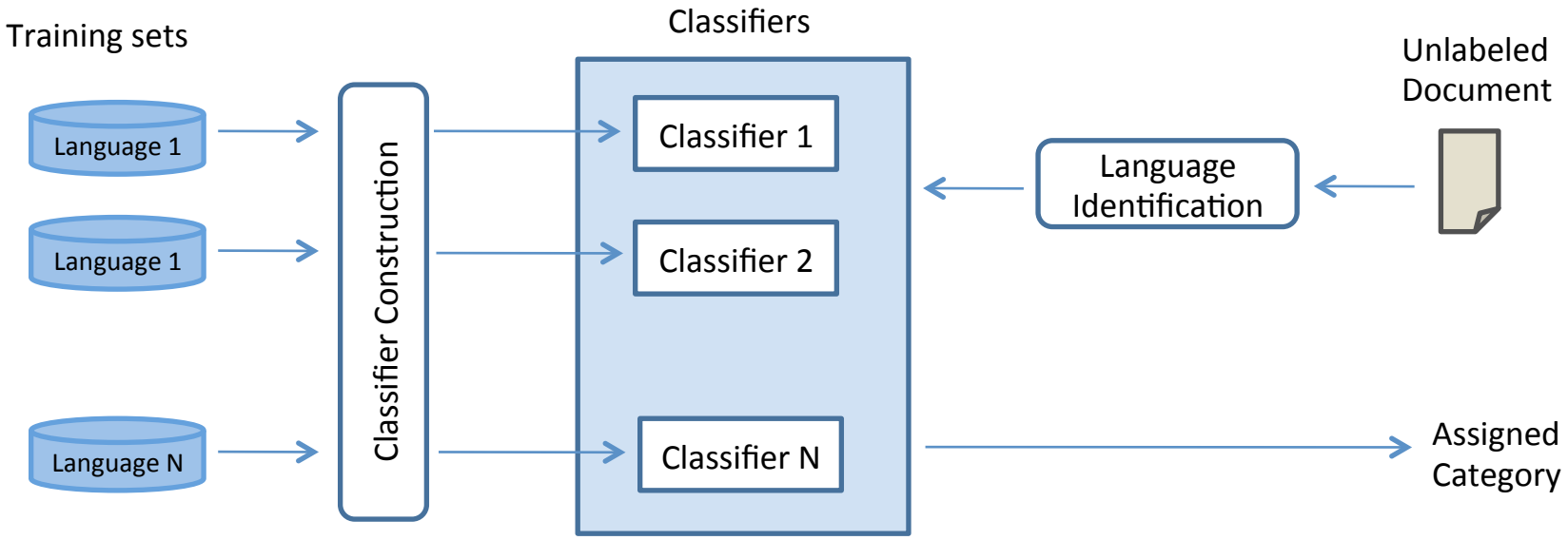
Possible applications?

Complicated or challenging situations?

Poly-lingual classification

- Two main steps:
  - Learning of categorization model(s) from a set of pre classified training documents written in different languages
  - Assignment of unclassified documents to predefined categories on the basis of the induced text categorization models
- The naïve approach considers the problem as multiple independent monolingual text categorization problems.
  - Architecture is a combination of several monolingual classifiers

General architecture



How to determine the language?

Problems with this architecture?

How to take advantage of resources from other languages?

Cross-lingual text classification

- It consists of using a labeled dataset in one language (L1) to classify unlabelled data in other language (L2).
- A method that is able to effectively perform this task would reduce the costs of building multi-language classification systems, since the human effort would be reduced to provide a training set in just one language.

How can we train a classifier of such characteristics?

How similar must be both document sets?

Using machine translation

- Main approach is to use translation to ensure that all documents are available in a single language
- Translation can be used in two different ways:
  - Training-Set Translation: the labeled set is translated into the target language(s).
    - *Became a poly-lingual approach*
  - Test-Set Translation: This approach consists in translating the unlabelled documents into one language (L1).

Which approach is better?

Problems of translation?

Problems caused by translations

- Certain drawbacks of the bag-of-words model become particularly severe in cross-lingual classification:
  - Spanish '*coche*' is generally mapped to '*car* ', whereas French '*voiture*' is translated to '*automobile*'.
  - Spanish '*Me duele la cabeza*' to '*It hurts the head to me*', which does not contain the word '*headache*'.
  - In Japanese and Chinese, there are separate words for older and younger sisters.

How to tackle these problems?

keyword translation

- Most methods consider the translation of the whole documents.
- But our representation is based on a SET of words
  - Order is not capture; moreover, no all words are included.

Is it really important to have a GOOD translation?

- In order to reduce translation errors some methods only approach the translation of *keywords*.
- A variant is to translate the sentences containing the N more important keywords.
  - The purpose is to give some *context* to the translation machine.

How to select the keywords of a document?

What are the main characteristics of a keyword?

Keyword extraction

- Keywords are the set of significant words in a document that give high-level description of its content.
  - They give clue about the its main idea
- Two main ideas for keyword extraction:
  - Frequent words are more important
  - Very common words (in the collection) are not relevant to characterize the content of a given document.

$$a_{ik} = f_{ik} * \log\left(\frac{N}{n_i}\right)$$

Frequency of word *i* in document *k*

Size of the whole collection

Number of documents having word *i*

Keyword extraction by term distribution

Keywords of a document appear
here and there in the document

- Extract important terms in documents applying the TF-IDF criterion.
- Examine the distribution characteristics of those candidate keywords.
- Select as document keywords the terms with great frequency and wide distribution

$$s_i^2 = \frac{1}{(f_i - 1)} \sum_j (l_{ij} - m_j)^2$$ where m_j is mean of relative location j.

Supervised keyword extraction

- Consider the keyword extraction as a classification problem: the purpose is to determine whether a word belong to the class of *keywords* or ordinary words
  - Assume that there is a training set that can be used to learn how to identify keywords and using the knowledge gained from the training set
- Some common used features are:
  - Frequency of the word in the document, inverse document frequency, position of the word in the document, position of the word according to the paragraph, format of the word, POS tag.

Other problems of CL text classification

- It is clear that, in spite of a perfect translation, there is also a *cultural distance* between both languages, which will inevitably affect the classification performance.
- As an example, consider the case of news about sports from France (in French) and from US (in English):
  - The first will include more documents about soccer, rugby and cricket
  - The later will mainly consider notes about baseball, basketball and American football.

How to address this issue?

An EM based algorithm for CLTC

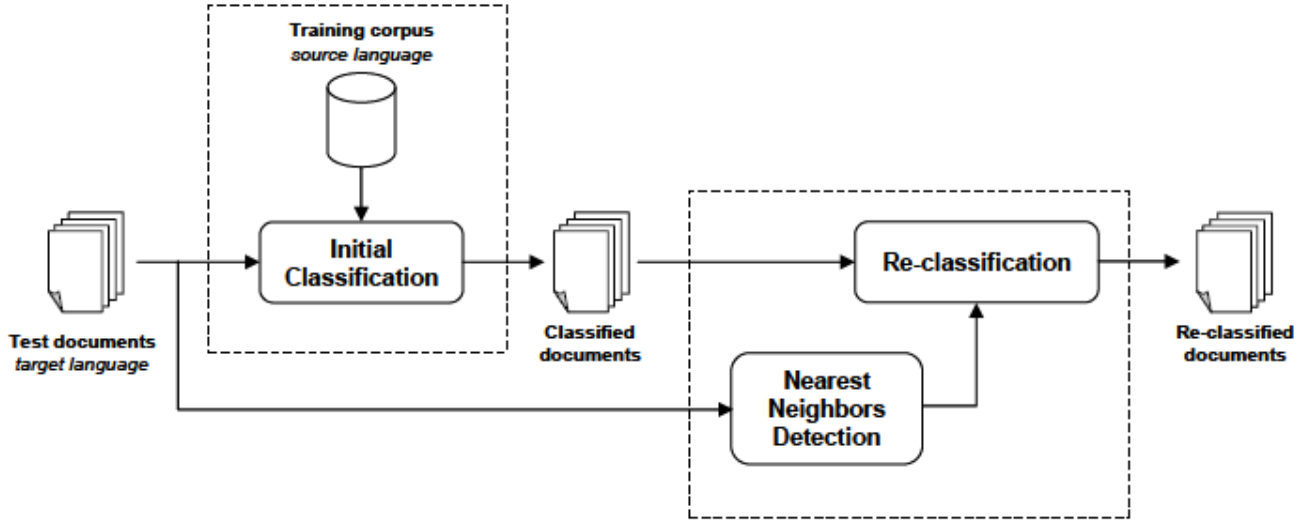
- Uses two different sets of data:
  - a set of manually labeled documents in language L_1
  - a *large* amount of unlabeled documents in the target language L_2.
- The main process:
  1. Translate training set to L2.
  2. Build a classifier using the labeled translated examples
  3. Use information in unlabeled examples from L2 to iteratively enrich the classifier
- The idea is that, even if the labels are not available, useful statistical properties can be extracted by looking at the distribution of terms in unlabeled texts.

Rigutini L., Maggini M., and Liu B. An EM based training algorithm for Cross-Language Text Categorization. 2005 IEEE/WIC/ACM International Conference on Web Intelligence. Compiegne, France, Sept. 2005.

Re-classification using neighbor´s information

- Post-processing method for CLTC
- Its purpose is to reduce the classification errors caused by the cultural distance between the two given languages
- It takes advantage from the synergy between similar documents from the target corpus in order to achieve their re-classification.
- It relies on the idea that similar documents from the target corpus are about the same topic, and, therefore, that they *must belong to the same category*.

Scheme of the method



- Iteratively, modify the current class of a document by considering information from their neighbors
  - If all neighbors belong to the same class, assign that class to the document
  - If neighbors do not belong to the same class, maintain current classification
  - Iterate σ times, or repeat until no document changes their category.

Results

Source language (Training set)	Target language (test set)	Vocabulary (training set)	Vocabulary (test set)	Vocabulary intersection	Percentage intersection (w.r.t test set)	Accuracy
English	English	10892	7658	5452	71%	0.917
Spanish	Spanish	12295	8051	5182	64%	0.917
French	French	14072	9258	6000	65%	0.933

Source language (Training set)	Target language (test set)	Initial Accuracy	Number of Neighbors 3	4	5
Translating training set to target language					
French	English	0.858	**0.958** (1)	0.925 (1)	0.925 (2)
Spanish	English	0.817	**0.900** (1)	**0.900** (2)	0.883 (3)
French	Spanish	0.833	**0.842** (1)	**0.842** (1)	**0.842** (1)
English	Spanish	0.717	0.725 (3)	**0.733** (4)	0.725 (1)
Spanish	French	0.808	**0.833** (1)	0.817 (1)	0.825 (1)
English	French	0.758	0.775 (1)	**0.767** (1)	**0.767** (1)
Translating test set to source language					
English	French	0.767	0.758 (2)	0.767 (1)	0.767 (1)
English	Spanish	0.750	0.750 (0)	0.750 (0)	0.750 (0)
Spanish	French	0.792	0.808 (1)	0.808 (1)	**0.817** (1)
Spanish	English	0.850	**0.908** (1)	0.892 (1)	0.892 (1)
French	Spanish	0.800	**0.817** (1)	0.808 (1)	**0.817** (1)
French	English	0.867	**0.925** (2)	0.892 (1)	0.892 (1)

Alternative: using a multilingual wordnet

- Instead of translating documents from one language to other, make them comparable by means of a multilingual wordnet.
- A wordnet is a large lexical database organized in terms of meanings.
  - Synonym words are grouped into synset ({car, auto, automobile, machine, motorcar})
- In a multilingual wordnet there are relations between related synsents
  - It is possible to go from the words in one language to similar words in any other language.

Using multilingual wordnets

- Idea is representing documents by a common (monolingual) set of concepts, and not by a common set of words.
- Advantages:
  - Synonym is captured (car and auto represented by the same instance)
  - Generalization is possible (if one document talk about lions, it somehow talk about felines)
- Disadvantages:
  - More difficult to have a multilingual wordnet than a translation system.
  - A BIG problem: word sense disambiguation

Alternative 2: Hybrid approach

1. Translate all documents to English
   - Training and test sets
   - Because English has the largest wordnet
2. Represent documents by a bag-of-synsets
3. Applied any supervised learning approach to learn from this representation.

Advantages:

- Not necessary to have/construct a wordnet for each language
- WSD in only one single language