Utility Theory, Minimum Effort, and Predictive Coding

Fabrizio Sebastiani
(Joint work with Giacomo Berardi and Andrea Esuli)

Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy

DESI V – Roma, IT, 14 June 2013

What I'll be talking about

- A talk about text classification ("predictive coding"), about humans in the loop, and about how to best support their work
- I will be looking at scenarios in which
  1. text classification technology is used for identifying documents belonging to a given class / relevant to a given query ...
  2. ... but the level of accuracy that can be obtained from the classifier is not considered sufficient ...
  3. ... with the consequence that one or more human assessors are asked to inspect (and correct where appropriate) a portion of the classification decisions, with the goal of increasing overall accuracy.
- How can we support / optimize the work of the human assessors?

What I'll be talking about

- A talk about text classification ("predictive coding"), about humans in the loop, and about how to best support their work
- I will be looking at scenarios in which
  1. text classification technology is used for identifying documents belonging to a given class / relevant to a given query ...
  2. ... but the level of accuracy that can be obtained from the classifier is not considered sufficient ...
  3. ... with the consequence that one or more human assessors are asked to inspect (and correct where appropriate) a portion of the classification decisions, with the goal of increasing overall accuracy.
- How can we support / optimize the work of the human assessors?

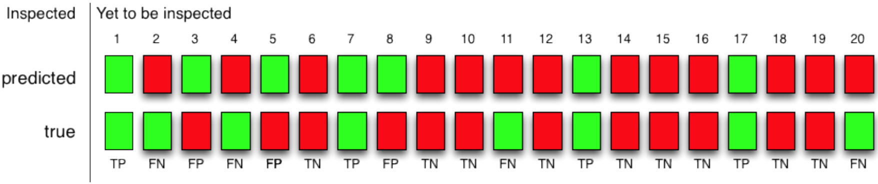
What I'll be talking about

- A talk about text classification ("predictive coding"), about humans in the loop, and about how to best support their work
- I will be looking at scenarios in which
  1. text classification technology is used for identifying documents belonging to a given class / relevant to a given query ...
  2. ... but the level of accuracy that can be obtained from the classifier is not considered sufficient ...
  3. ... with the consequence that one or more human assessors are asked to inspect (and correct where appropriate) a portion of the classification decisions, with the goal of increasing overall accuracy.
- How can we support / optimize the work of the human assessors?

A worked out example

		predicted	
		Y	**N**
true	**Y**	$TP = 4$	$FP = 3$
	N	$FN = 4$	$TN = 9$

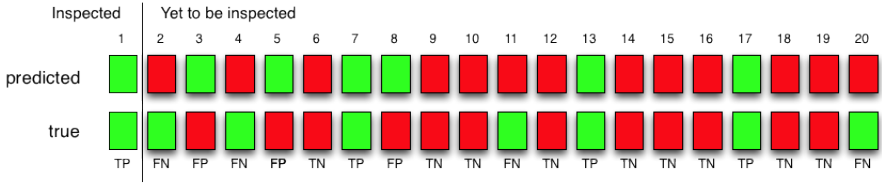
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.53$$



A worked out example (cont'd)

		predicted Y	predicted N
true	Y	$TP = 4$	$FP = 3$
	N	$FN = 4$	$TN = 9$

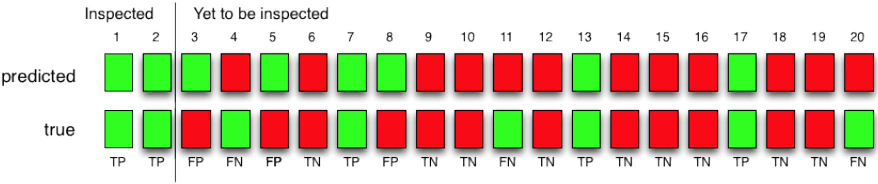
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.53$$



A worked out example (cont'd)

		predicted	
		Y	**N**
true	**Y**	$TP = 5$	$FP = 3$
	N	$FN = 3$	$TN = 9$

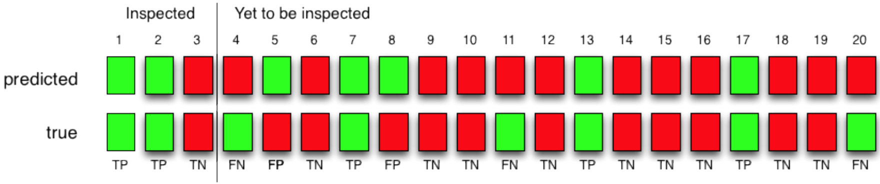
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.63$$



A worked out example (cont'd)

		predicted	
		Y	N
true	Y	$TP = 5$	$FP = 2$
	N	$FN = 3$	$TN = 10$

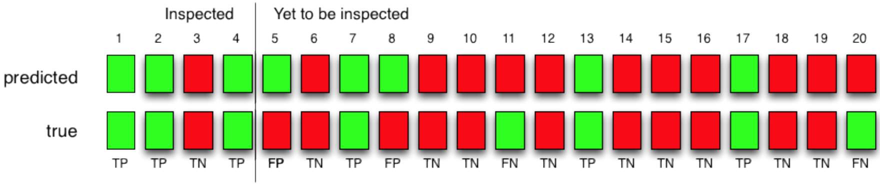
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.67$$



A worked out example (cont'd)

		predicted	
		Y	**N**
true	**Y**	$TP = 6$	$FP = 2$
	N	$FN = 2$	$TN = 10$

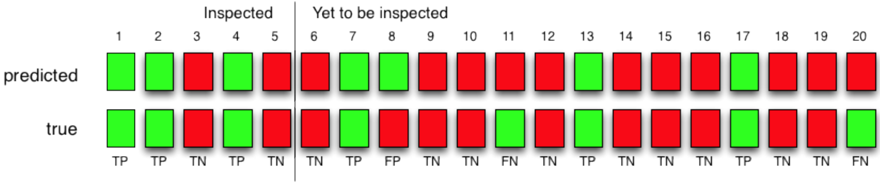
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.75$$



A worked out example (cont'd)

		predicted	
		Y	**N**
true	**Y**	$TP = 6$	$FP = 1$
	N	$FN = 2$	$TN = 11$

$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.80$$



What I'll be talking about (cont'd)

- We need methods that
  - given a desired level of accuracy, minimize the assessors' effort necessary to achieve it; alternatively,
  - given an available amount of human assessors' effort, maximize the accuracy that can be obtained through it
- This can be achieved by ranking the automatically classified documents in such a way that, by starting the inspection from the top of the ranking, the cost-effectiveness of the annotators' work is maximized
- We call the task of generating such a ranking Semi-Automatic Text Classification (SATC)

What I'll be talking about (cont'd)

- We need methods that
  - given a desired level of accuracy, minimize the assessors' effort necessary to achieve it; alternatively,
  - given an available amount of human assessors' effort, maximize the accuracy that can be obtained through it
- This can be achieved by ranking the automatically classified documents in such a way that, by starting the inspection from the top of the ranking, the cost-effectiveness of the annotators' work is maximized
- We call the task of generating such a ranking Semi-Automatic Text Classification (SATC)

What I'll be talking about (cont'd)

- Previous work has addressed SATC via techniques developed for "active learning"
- In both cases, the automatically classified documents are ranked with the goal of having the human annotator start inspecting/correcting from the top; however
  - in active learning the goal is providing new training examples
  - in SATC the goal is increasing the overall accuracy of the classified set
- We claim that a ranking generated "à la active learning" is suboptimal for SATC[1]

[1] G Berardi, A Esuli, F Sebastiani. A Utility-Theoretic Ranking Method for Semi-Automated Text Classification. Proceedings of the 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012), Portland, US, 2012.

What I'll be talking about (cont'd)

- Previous work has addressed SATC via techniques developed for "active learning"
- In both cases, the automatically classified documents are ranked with the goal of having the human annotator start inspecting/correcting from the top; however
  - in active learning the goal is providing new training examples
  - in SATC the goal is increasing the overall accuracy of the classified set
- We claim that a ranking generated "à la active learning" is suboptimal for SATC[1]

[1] G Berardi, A Esuli, F Sebastiani. A Utility-Theoretic Ranking Method for Semi-Automated Text Classification. Proceedings of the 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012), Portland, US, 2012.

Outline of this talk

1. We discuss how to measure "error reduction" (i.e., increase in accuracy)
2. We discuss a method for maximizing the expected error reduction for a fixed amount of annotation effort
3. We show some promising experimental results

Outline

1. Error Reduction, and How to Measure it
2. Error Reduction, and How to Maximize it
3. Some Experimental Results

Error Reduction, and how to measure it

Assume we have

1. class (or “query”) c;
2. classifier h for c;
3. set of unlabeled documents D that we have automatically classified by means of h, so that every document in D is associated
   - with a binary decision (Y or N)
   - with a confidence score (a positive real number)
4. measure of accuracy A, ranging on [0,1]

Error Reduction, and how to Measure it (cont'd)

- We will assume that A is

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{(2 \cdot TP) + FP + FN}$$

  but any "set-based" measure of accuracy (i.e., based on a contingency table) may be used
- An amount of error, measured as $E = (1 - A)$, is present in the automatically classified set D
- Human annotators inspect-and-correct a portion of D with the goal of reducing the error present in D

Error Reduction, and how to Measure it (cont'd)

- We will assume that A is

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{(2 \cdot TP) + FP + FN}$$

  but any "set-based" measure of accuracy (i.e., based on a contingency table) may be used
- An amount of error, measured as $E = (1 - A)$, is present in the automatically classified set D
- Human annotators inspect-and-correct a portion of D with the goal of reducing the error present in D

Error Reduction, and how to Measure it (cont'd)

- We define error at rank n (noted as $E(n)$) as the error still present in D after the annotator has inspected the documents at the first n rank positions
  - $E(0)$ is the initial error generated by the automated classifier
  - $E(|D|)$ is 0
- We define error reduction at rank n (noted as $ER(n)$) to be

$$ER(n) = \frac{E(0) - E(n)}{E(0)}$$

  the error reduction obtained by the annotator who inspects the docs at the first n rank positions
  - $ER(n) \in [0, 1]$
  - $ER(n) = 0$ indicates no reduction
  - $ER(n) = 1$ indicates total elimination of error

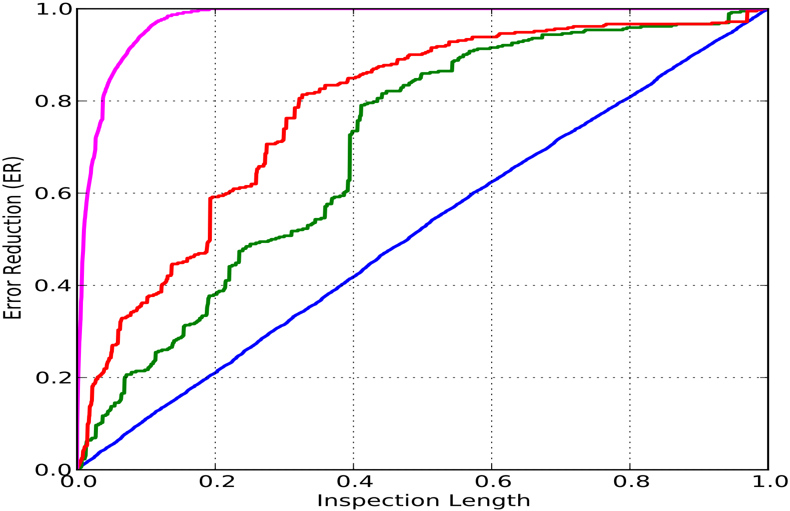
Error Reduction, and how to Measure it (cont'd)

- We define error at rank n (noted as $E(n)$) as the error still present in D after the annotator has inspected the documents at the first n rank positions
  - $E(0)$ is the initial error generated by the automated classifier
  - $E(|D|)$ is 0
- We define error reduction at rank n (noted as $ER(n)$) to be

$$ER(n) = \frac{E(0) - E(n)}{E(0)}$$

  the error reduction obtained by the annotator who inspects the docs at the first n rank positions
  - $ER(n) \in [0, 1]$
  - $ER(n) = 0$ indicates no reduction
  - $ER(n) = 1$ indicates total elimination of error

Error Reduction, and how to Measure it (cont'd)



Outline

Error Reduction, and how to Maximize it

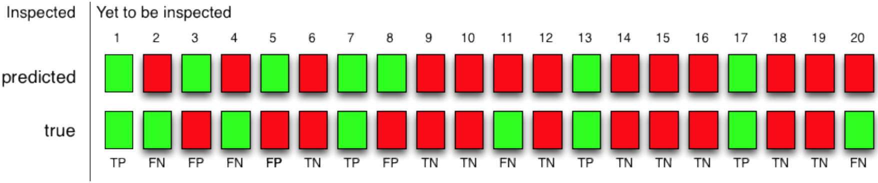
Problem

How should we rank the documents in D so as to maximize the expected error reduction?

A worked out example

		predicted	
		Y	**N**
true	**Y**	$TP = 4$	$FP = 3$
	N	$FN = 4$	$TN = 9$

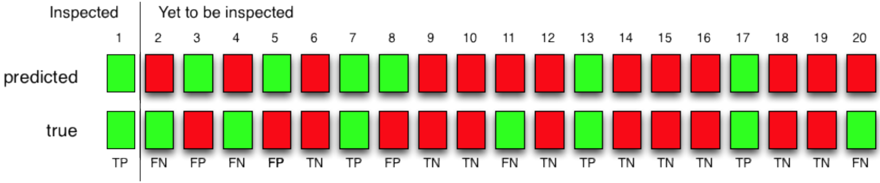
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.53$$



A worked out example (cont'd)

		predicted Y	predicted N
true	Y	$TP = 4$	$FP = 3$
	N	$FN = 4$	$TN = 9$

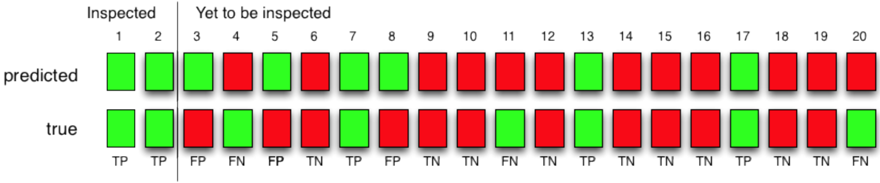
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.53$$



A worked out example (cont'd)

		predicted Y	predicted N
true	Y	$TP = 5$	$FP = 3$
	N	$FN = 3$	$TN = 9$

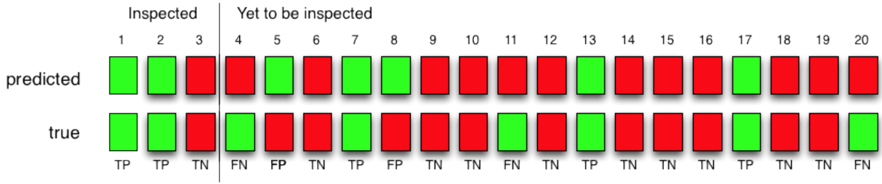
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.63$$



A worked out example (cont'd)

		predicted	
		Y	**N**
true	**Y**	$TP = 5$	$FP = 2$
	N	$FN = 3$	$TN = 10$

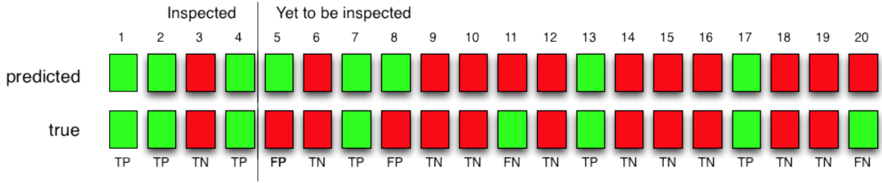
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.67$$



A worked out example (cont'd)

		predicted Y	predicted N
true	Y	$TP = 6$	$FP = 2$
	N	$FN = 2$	$TN = 10$

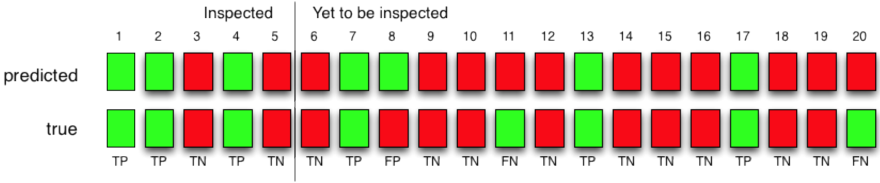
$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.75$$



A worked out example (cont'd)

		predicted Y	predicted N
true	Y	$TP = 6$	$FP = 1$
true	N	$FN = 2$	$TN = 11$

$$F_1 = \frac{2TP}{2TP + FP + FN} = 0.80$$



Error Reduction, and how to Maximize it

- Problem: how should we rank the documents in D so as to maximize the expected error reduction?
- Intuition 1: Documents that have a higher probability of being misclassified should be ranked higher
- Intuition 2: Documents that, if corrected, bring about a higher gain (i.e., a bigger impact on A) should be ranked higher
  - Here, consider that a false positive and a false negative may have different impacts on A (e.g., when $A \equiv F_\beta$, for any value of β)

Bottom line

Documents that have a higher utility ($=$ probability $\times$ gain) should be ranked higher

Error Reduction, and how to Maximize it

- Problem: how should we rank the documents in D so as to maximize the expected error reduction?
- Intuition 1: Documents that have a higher probability of being misclassified should be ranked higher
- Intuition 2: Documents that, if corrected, bring about a higher gain (i.e., a bigger impact on A) should be ranked higher
    - Here, consider that a false positive and a false negative may have different impacts on A (e.g., when $A \equiv F_\beta$, for any value of β)

Bottom line

Documents that have a higher utility (= probability $\times$ gain) should be ranked higher

Error Reduction, and how to Maximize it (cont'd)

- Given a set Ω of mutually disjoint events, a utility function is defined as

$$U(\Omega) = \sum_{\omega \in \Omega} P(\omega) G(\omega)$$

  where

  - $P(\omega)$ is the probability of occurrence of event ω
  - $G(\omega)$ is the gain obtained if event ω occurs

- We can thus estimate the utility, for the aims of increasing A, of manually inspecting a document d as

$$U(TP, TN, FP, FN) = P(FP) \cdot G(FP) + P(FN) \cdot G(FN)$$

  provided we can estimate

  - If d is labelled with class c: $P(FP)$ and $G(FP)$
  - If d is not labelled with class c: $P(FN)$ and $G(FN)$

Error Reduction, and how to Maximize it (cont'd)

- Given a set Ω of mutually disjoint events, a utility function is defined as

$$U(\Omega) = \sum_{\omega \in \Omega} P(\omega) G(\omega)$$

  where
  - $P(\omega)$ is the probability of occurrence of event ω
  - $G(\omega)$ is the gain obtained if event ω occurs

- We can thus estimate the utility, for the aims of increasing A, of manually inspecting a document d as

$$U(TP, TN, FP, FN) = P(FP) \cdot G(FP) + P(FN) \cdot G(FN)$$

  provided we can estimate
  - If d is labelled with class c: $P(FP)$ and $G(FP)$
  - If d is not labelled with class c: $P(FN)$ and $G(FN)$

Error Reduction, and how to Maximize it (cont'd)

- Estimating $P(FP)$ and $P(FN)$ (the probability of misclassification) can be done by converting the confidence score returned by the classifier into a probability of correct classification
    - Tricky: requires probability "calibration" via a generalized sigmoid function to be optimized via k-fold cross-validation
- Gains $G(FP)$ and $G(FN)$ can be defined "differentially"; i.e.,
    - The gain obtained by correcting a FN is $(A^{FN \rightarrow TP} - A)$
    - The gain obtained by correcting a FP is $(A^{FP \rightarrow TN} - A)$
    - Gains need to be estimated by estimating the contingency table on the training set via k-fold cross-validation
    - Key observation: in general, $G(FP) \neq G(FN)$

Error Reduction, and how to Maximize it (cont'd)

- Estimating $P(FP)$ and $P(FN)$ (the probability of misclassification) can be done by converting the confidence score returned by the classifier into a probability of correct classification
  - Tricky: requires probability "calibration" via a generalized sigmoid function to be optimized via k-fold cross-validation
- Gains $G(FP)$ and $G(FN)$ can be defined "differentially"; i.e.,
  - The gain obtained by correcting a FN is $(A^{FN \rightarrow TP} - A)$
  - The gain obtained by correcting a FP is $(A^{FP \rightarrow TN} - A)$
  - Gains need to be estimated by estimating the contingency table on the training set via k-fold cross-validation
  - Key observation: in general, $G(FP) \neq G(FN)$

Error Reduction, and how to Maximize it (cont'd)

- Estimating $P(FP)$ and $P(FN)$ (the probability of misclassification) can be done by converting the confidence score returned by the classifier into a probability of correct classification
  - Tricky: requires probability "calibration" via a generalized sigmoid function to be optimized via k-fold cross-validation
- Gains $G(FP)$ and $G(FN)$ can be defined "differentially"; i.e.,
  - The gain obtained by correcting a FN is $(A^{FN \rightarrow TP} - A)$
  - The gain obtained by correcting a FP is $(A^{FP \rightarrow TN} - A)$
  - Gains need to be estimated by estimating the contingency table on the training set via k-fold cross-validation
  - Key observation: in general, $G(FP) \neq G(FN)$

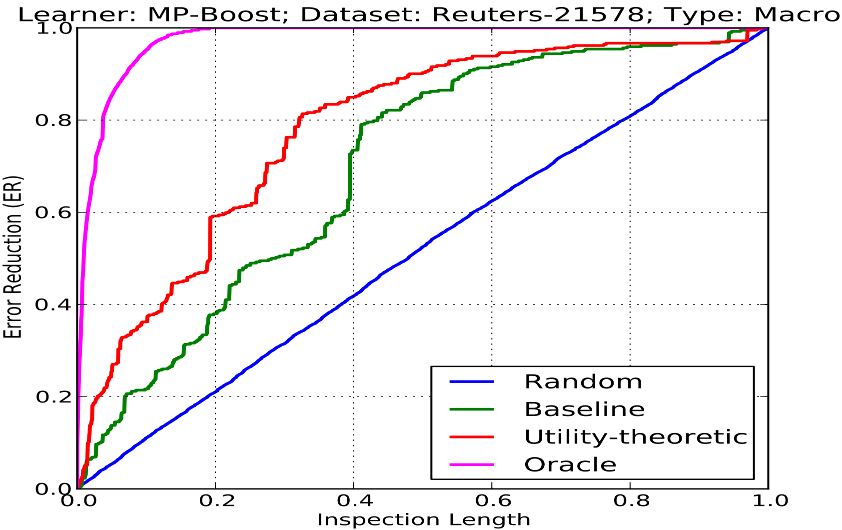
Outline

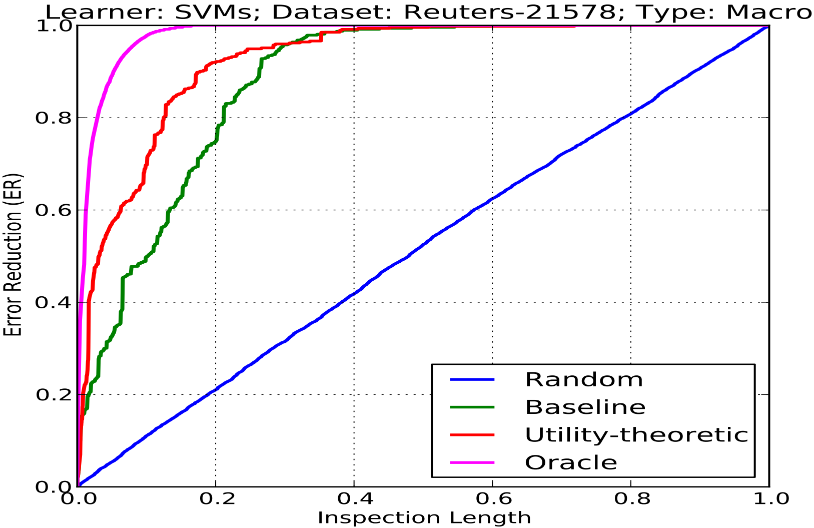
Some Experimental Results

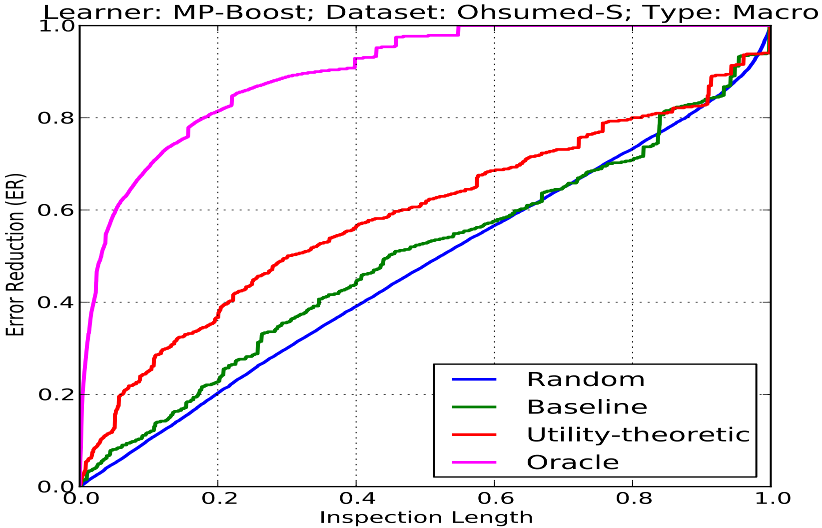
- Learning algorithms: MP-BOOST, SVMs
- Datasets:

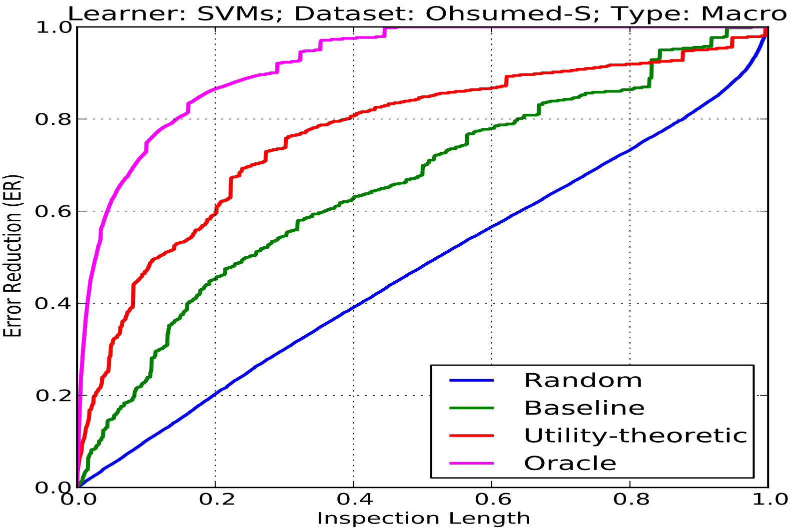
	# Cats	# Training	# Test	F_1^M MP-BOOST	F_1^M SVMs
REUTERS-21578	115	9603	3299	0.608	0.527
OHSUMED-S	97	12358	3652	0.479	0.478

- Baseline: ranking by probability of misclassification, equivalent to applying our ranking method with $G(FP) = G(FN) = 1$


Learner: MP-Boost; Dataset: Reuters-21578; Type: Macro
1.0
0.8
0.6
0.4
0.2
0.0
Error Reduction (ER)
0.0
0.2
0.4
0.6
0.8
1.0
Inspection Length
Random
Baseline
Utility-theoretic
Oracle


Learner: SVMs; Dataset: Reuters-21578; Type: Macro
Error Reduction (ER)
Inspection Length
0.0
0.2
0.4
0.6
0.8
1.0
Random
Baseline
Utility-theoretic
Oracle


Learner: MP-Boost; Dataset: Ohsumed-S; Type: Macro
Error Reduction (ER)
Inspection Length
0.0
0.2
0.4
0.6
0.8
1.0
Random
Baseline
Utility-theoretic
Oracle


Learner: SVMs; Dataset: Ohsumed-S; Type: Macro
Error Reduction (ER)
Inspection Length
0.0
0.2
0.4
0.6
0.8
1.0
Random
Baseline
Utility-theoretic
Oracle

A few side notes

- This approach allows the human annotator to know, at any stage of the inspection process, what the estimated accuracy is at that stage
  - Estimate accuracy at the beginning of the process, via k-fold cross validation
  - Update after each correction is made
- This approach lends itself to having more than one assessor working in parallel on the same inspection task
- Recent research I have not discussed today :
  - A "dynamic" SATC method in which gains are updated after each correction is performed
  - "Microaveraging" and "Macroaveraging" -oriented methods

Concluding Remarks

- Take-away message: Semi-automatic text classification needs to be addressed as a task in its own right
  - Active learning typically makes use of probabilities of misclassification but does not make use of gains ⇒ ranking "à la active learning" is suboptimal for SATC
- The use of utility theory means that the ranking algorithm is optimized for a specific accuracy measure ⇒ Choose the accuracy measure the best mirrors your applicative needs (e.g., F_β with $\beta > 1$), and choose it well!
- SATC is important, since in more and more application contexts the accuracy obtainable via completely automatic text classification is not sufficient; more and more frequently humans will need to enter the loop

Thank you!