Sentence-Level Confidence Estimation for MT

Lucia Specia

Nicola Cancedda, Marc Dymetman, Craig Saunders - Xerox

Marco Turchi, Nello Cristianini – Univ. Bristol

Zhuoran Wang, John Shawe-Taylor – UCL

Outline

- The task of Confidence Estimation for Machine Translation
- Approach
  - Features
  - Algorithms
  - Method
- Experiments
  - **Scenario 1**: providing score to the end-user that is as close as possible to a human score
  - **Scenario 2**: filtering out bad translations for professional translators

The task of CE for MT

- **Goal**: given the output of a Machine Translation (MT) system for a given input, provide an estimate of its **quality**.

- **Motivation:** assessing the quality of translations is
- **Time consuming**:

  Los investigadores dicen gripe porcina tiene «pleno potencial pandémico", difundiendo rápidamente entre las personas y es probable que vaya mundial en los próximos seis a nueve meses, con uno de cada tres de la población mundial infectada.

- **Not possible** - if user does not know the source language:

  शोधकर्ताओं सूअर फ्लू पूर्ण पैन्डेमिक की क्षमता है , लोगों को तत्काल और वैश्विक अगले छह से नौ महीनों में , एक तीन दुनिया की आबादी ....

xerox

The task of CE for MT

- **Uses:**

Is it worth providing this translation as suggestion to the professional translator?

– **Filter** out “bad” translations to avoid **professional translators** wasting time reading / post-editing them.

Should this translation be highlighted as “suspect” to the reader?

– Make **end-users aware** of the translation quality.

General approach

- **Different from MT evaluation (BLEU, NIST**, etc**)**: reference translations are **NOT** available
- **Unit**: word, phrase or sentence
- **Embedded** to SMT system (word or phrase probabilities) or **dedicated** layer (machine learning problem)
- **Traditional** approach**:**
  - **Binary problem**: distinguish between **good** and **bad** translations
- **Training data**: data automatically annotated with NIST/BLEU or manually annotated (e.g. 1-5 scores)

General approach

- **Different from MT evaluation (BLEU, NIST**, etc**)**: reference translations are **NOT** available
- **Unit**: word, phrase or sentence
- **Embedded** to SMT system (word or phrase probabilities) or **dedicated** layer (machine learning problem)
- **Traditional** approach**:**
  - **Binary problem**: distinguish between **good** and **bad** translations
  - **Continuous score**
- **Training data**: data automatically annotated with NIST/BLEU or manually annotated (e.g. 1-5 scores), from several **MT systems**, text **domains** and **language pairs**

Method

- **Identify and extract information sources**.
- **Refine the set of information sources** to keep only the relevant ones
  - Increase **performance**
- **Learn a model to produce quality scores**
  - **Regression algorithm**
- **Apply the model to predict quality scores** for new translations

Features

- **Resource & language independent** features
- **Black-box (77):** from the **input** and **translation sentences**, monolingual or parallel **corpora**, e.g.:
  - Source and target sentence lengths and their ratios
  - Language model and other statistics in the corpus
  - Shallow syntax checking (target and target against source)
  - Average number of possible translations per source word (SMT)
- **Practical scenario**:
  - Useful when it is not possible to have **access to internal features** of the MT systems (commercial systems, e.g.).
  - Provides a way to perform the task of CE **across different MT systems**, which may use different frameworks.

Features

- **Glass-box (54)**: depend on some aspect of the **translation process**, e.g.:
  - Language model (target) using n-best list – word/phrase-based
  - Proximity with other hypothesis in the n-best list
  - MT base model features
    - Distortion count, gap count, (compositional) bi-phrase probability
  - Search nodes in the graph (aborted, pruned)
  - Proportion of unknown words in the source

- **Richer scenario**:
  - When it is possible to have **access to internal features** of the MT systems.

Learning methods

- **Feature selection:** Partial Least Squares (PLS)
- **Regression:** PLS, SVM

Partial Least Square Regression

- Projects the original data onto a different space of latent variables (or "**components**")
- Used to find the fundamental relations between two matrices (input **X** and **Y** response variables): tries to find the multidimensional direction in the X space that explains the maximum variance direction in the Y space
  - Provides by-product an ordering of the original features according to their importance
- Particularly indicated when the features in X are strongly correlated (**multicollinearity**) → the case in our datasets

Partial Least Square Regression

- Ordinary multiple regression problem:

$$Y = XB_w + F$$

- Where:
  - **B_w** is the regression matrix computed directly using an optimal number of components.
  - **F** is the residual matrix.
  - When X is standardized, an element of **B_w** with large absolute value indicates an **important X-variable**.

Feature Selection with PLS

- Method:
  - Compute the B_w matrix on some training data for different numbers of components (all possible)
  - Sort the absolute value of the b_w-coefficients. This produces a list of features from the most important to the less important (L_b)
  - Select the top *n* features (and number of components) on a validation set
  - Produce predictions using these *n* features on a test set
  - Evaluate predictions using appropriate metrics

Feature Selection with PLS

- Method:
  - Compute the B_w matrix on some training data for different numbers of components (all possible)
  - Sort the absolute value of the b_w-coefficients. This produces a list of features from the most important to the less important (L_b)
  - Done for each *i-th* training subsample: obtain several $L_b(i)$

66	7	…	35	10
44	56	…	9	10
…	…	…	…	…
66	56		35	9

  - The final list L is obtained picking the most “voted” features for each column (mode): L = {66, 56, …, 35, 10}

Feature Selection with PLS

- Method:
  - Compute the B_w matrix on some training data for different numbers of components (all possible)
  - Sort the absolute value of the b_w-coefficients. This produces a list of features from the most important to the less important (L_b)
  - Select the top *n* features (and number of components) on a validation set
  - Add one feature one by one
  - Analyze learning curves to verify the prediction error
  - Select the top *n* features and the number of components that minimize the prediction error

Feature Selection with PLS

- Method:
  - Compute the B_w matrix on some training data for different numbers of components (all possible)
  - Sort the absolute value of the b_w-coefficients. This produces a list of features from the most important to the less important (L_b)
  - Select the top *n* features (and number of components) on a validation set
  - Produce predictions using these *n* features on a test set
    - **PLS** for regression

Feature Selection with PLS

- Method:
  - Compute the B_w matrix on some training data for different numbers of components (all possible)
  - Sort the absolute value of the b_w-coefficients. This produces a list of features from the most important to the less important (L_b)
  - Select the top *n* features on a validation set
  - Produce predictions using these *n* features on a test set
  - Evaluate predictions using appropriate metrics
    - Root Mean Square Error (**RMSPE**) over all subsamples

$$RMSPE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2}$$

Experiments – scenario 1

- WMT-2008 **Europarl English-Spanish** dev and test data
  - **4K** Translations: **SMT systems** trained on **1.4M** parallel sentences: **Matrax, Portage, Sinuhe and MMR (P-ES-1, P-ES-2, P-ES-3 and P-ES-4).**
  - Quality score: **1-4**

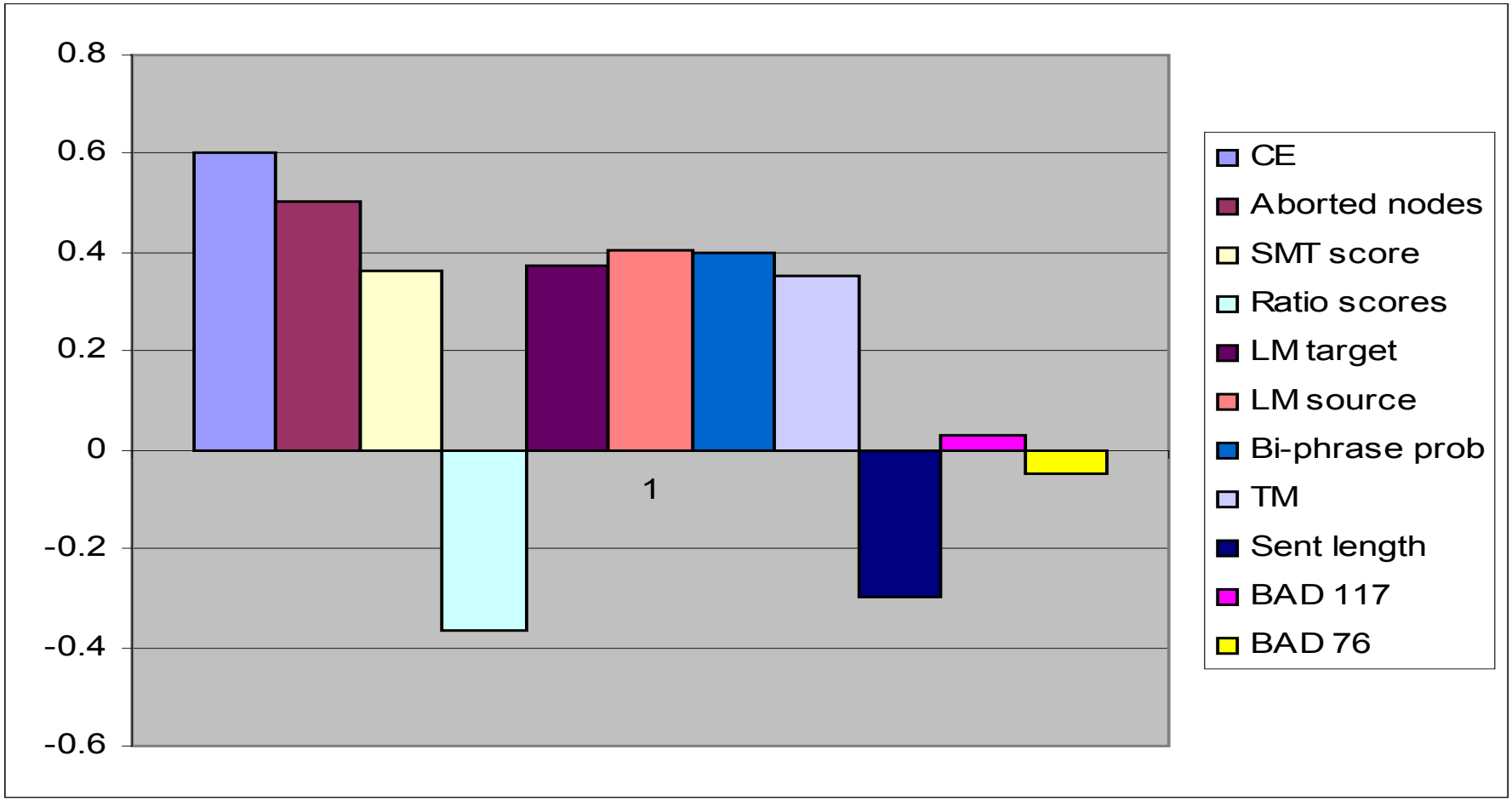
1: requires complete retranslation	2: editing quicker than retranslation
3: a little post editing needed	4: fit for purpose

- **Features**: Matrax (131), others 77 black-box.

MT	RMSPE	RMSPE all features
P-ES-1gb	0.690 ± 0.052	0.780 ± 0.385
P-ES-1	0.706 ± 0.059	0.793 ± 0.643
P-ES-2	0.653 ± 0.114	0.750 ± 0.541
P-ES-3	0.718 ± 0.144	0.745 ± 0.287
P-ES-4	0.603 ± 0.262	1.550 ± 3.551

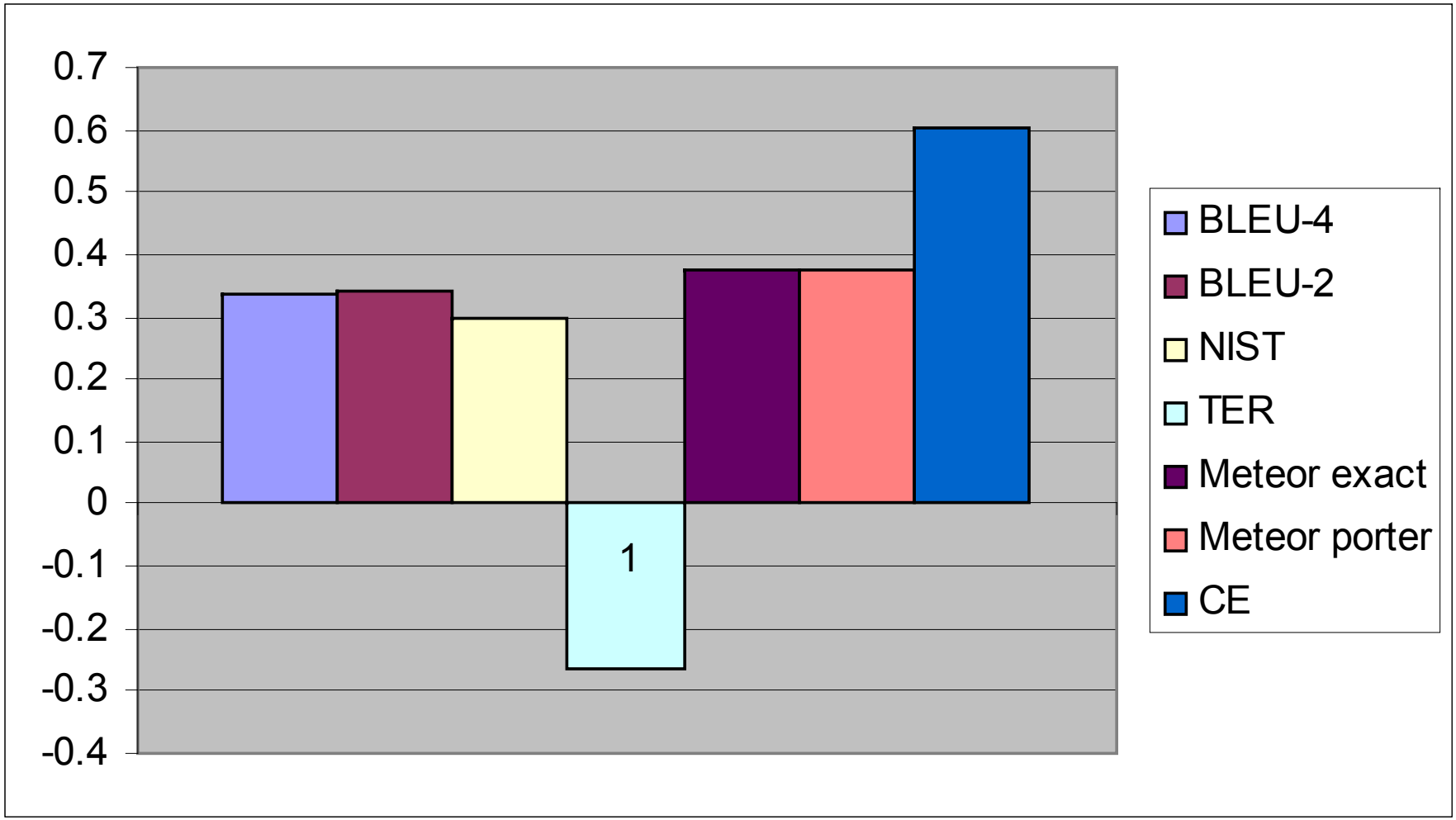
Experiments – scenario 1

- CE x best features (Pearson's correlation):



Experiments – scenario 1

- CE score x MT metrics: Pearson's correlation:



xerox

Experiments – scenario 1

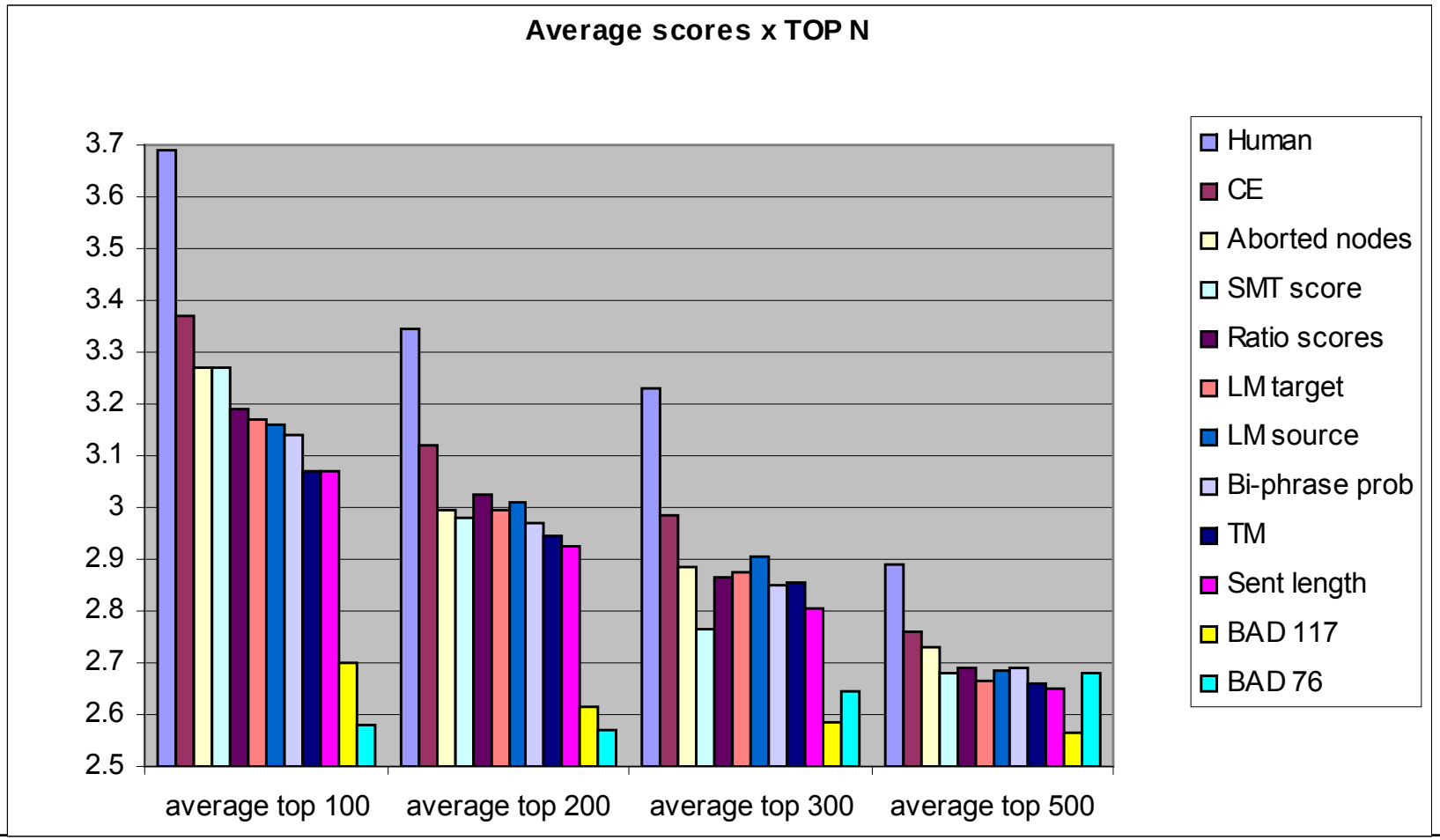
- CE score x MT metrics: Pearson's correlation across datasets (MT systems, language-pairs and text domains):

Test set	BLEU	NIST	TER	Meteor	Training set	CE score
en-es Matrax	0.296	0.254	-0.268	0.337	en-es Matrax	**0.542**
					en-es Portage	**0.478**
					en-es Sinuhe	**0.517**
					en-es MMR	**0.423**
					en-dk Matrax	**0.390**
en-es Sinuhe	0.209	0.195	-0.168	0.240	en-es Matrax	**0.547**
					en-es Portage	**0.531**
					en-es Sinuhe	**0.562**
					en-es MMR	**0.442**
					en-dk Matrax	**0.400**

xerox

Experiments – scenario 2

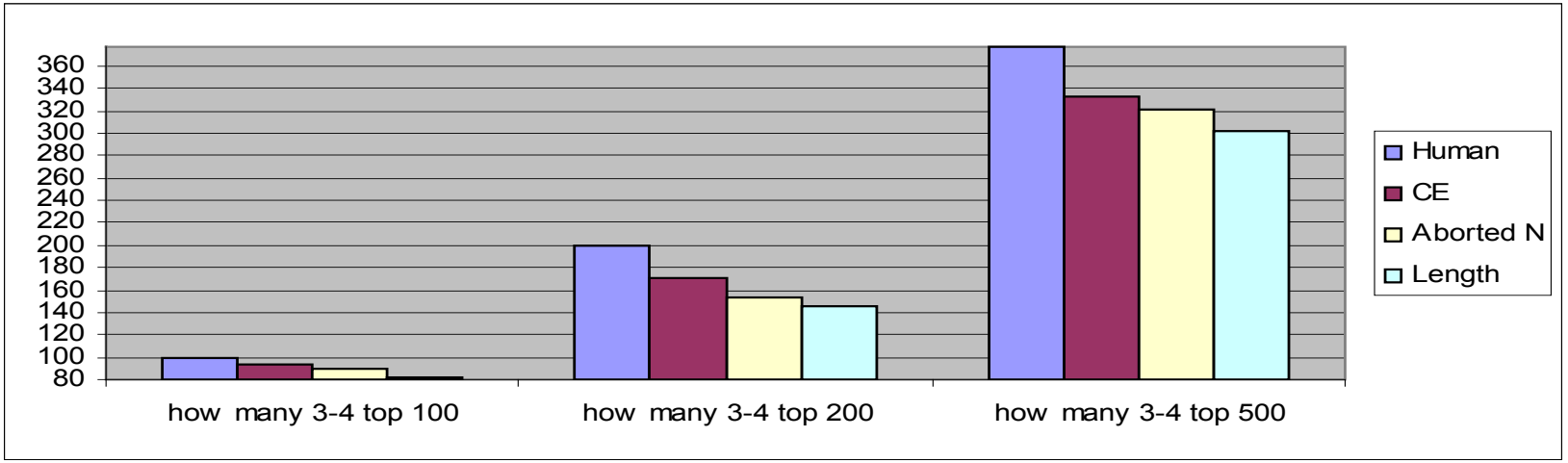
- Filter out bad translations (1-2) for professional translators
  - Average human scores in the top *n* translations:

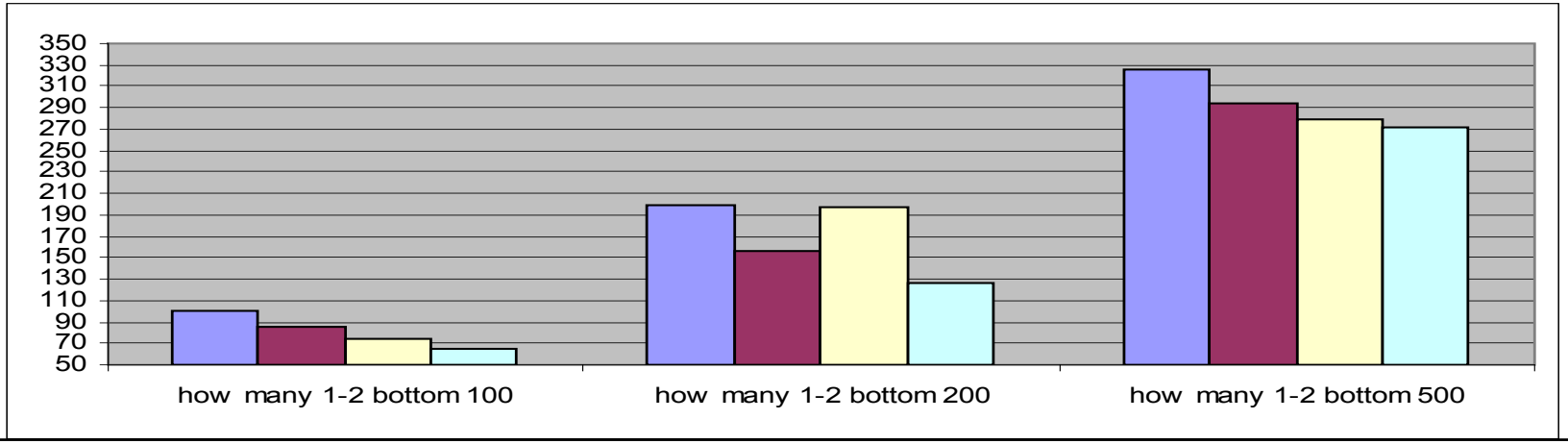


xerox

Experiments – scenario 2

- Filter out bad translations
  - Number of good (bad) translations in the top (bottom) *n* translations:





xerox

Inductive Confidence Machines

- **Scenario 2: better way to use the CE score for filtering out bad translations**
  - Use a technique to dynamically identify the **threshold** in the continuous score for **bad/good** translations
    - Controls the balance between **precision** and **recall** based on some expected **confidence level**
    - Higher **confidence level** → **higher precision**
      - In our scenario: **precision** is more relevant: guarantee that sentences selected as 'good' are indeed good - minimize **false positives**
- Technique:
  - Conformal prediction [Vovk, Gammerman & Shafer 2005]
    - Inductive Confidence Machine (ICM) [Papadopoulos et al. 2002]

Inductive Confidence Machines

- Given a pre-defined **confidence level** $1-\delta$, predict a region:

$$\{y : p(y) > \delta\}$$

- Split a training set $\{(x_1,y_1),\ldots,(x_l,y_l)\}$ of l examples into:
  - A proper training set $\{(x_1,y_1),\ldots,(x_m,y_m)\}$ with $m<l$ elements
  - A calibration set $\{(x_{m+1},y_{m+1}),\ldots,(x_l,y_l)\}$ with $k:=l-m$ elements

- Train a standard regression model on the proper training set

- Apply it the calibration set, and define a strangeness measure:

$$\alpha_i := \Delta(\hat{y}_{m+i}, y_{m+i}), \quad i = 1,\ldots,k.$$

- The p-value associated with a potential label y_{k+1} is:

$$p(y_{k+1}) := \frac{\#\{i = 1,\ldots,k : \alpha_i \geq \alpha_{k+1}\}}{k+1}.$$

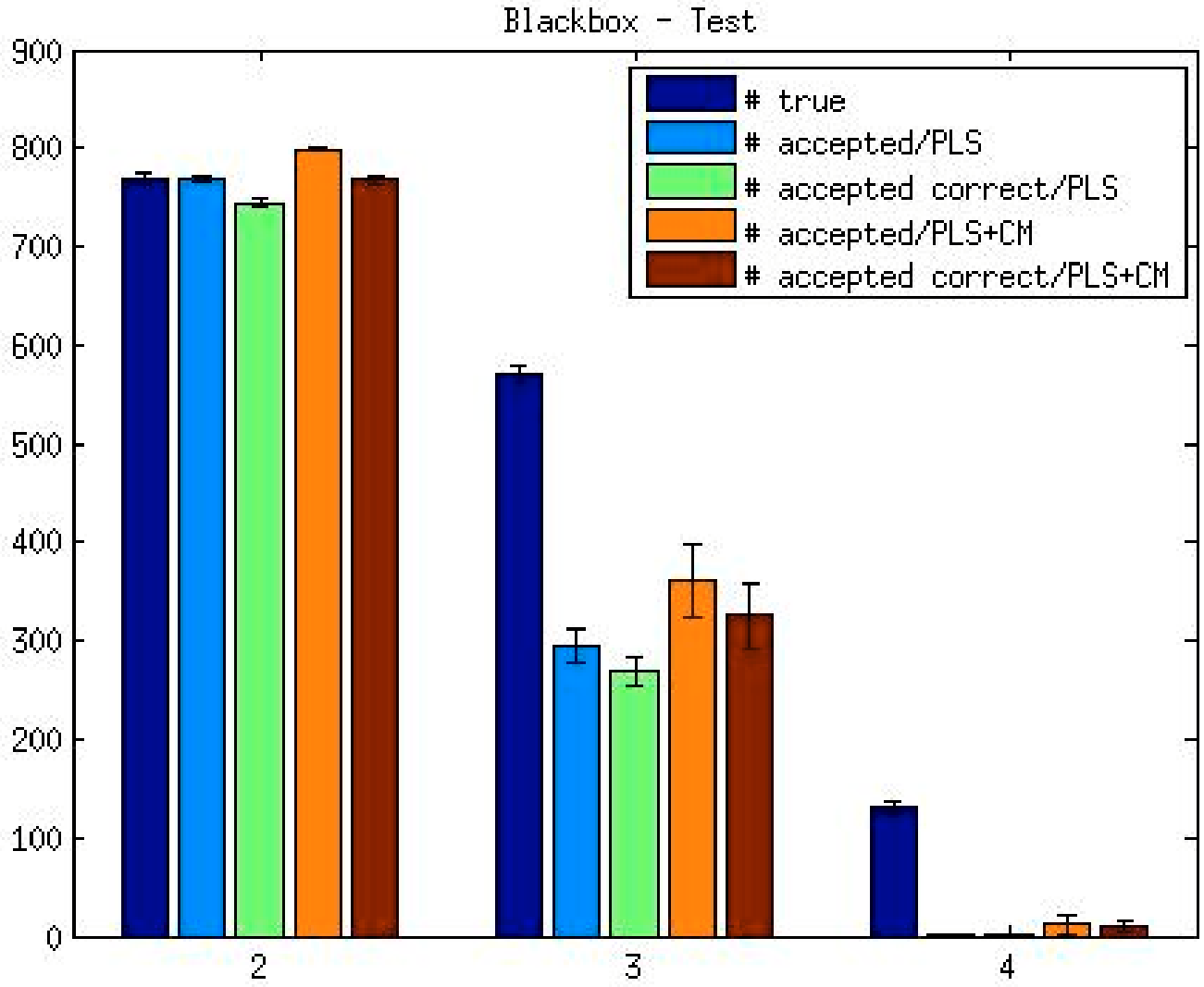
Inductive Confidence Machines

- Establishing the expected precision:
  - Confidence level $1\text{-}\delta$: expected precision
  - Search for a confidence threshold ρ of the regression predictions
    - A certain percent (e.g. 90%) of the predictions that $\geq \rho$ will have their true scores $\geq \tau$
  - For a fixed ρ, only consider those examples $y^* = f(x^*) \geq \rho$
  - Strangeness measure:
  - Binary search:

$$\alpha_i := \operatorname{sgn}(\tau - y_{m+i}) \cdot (\hat{y}_{m+i} - \rho)$$

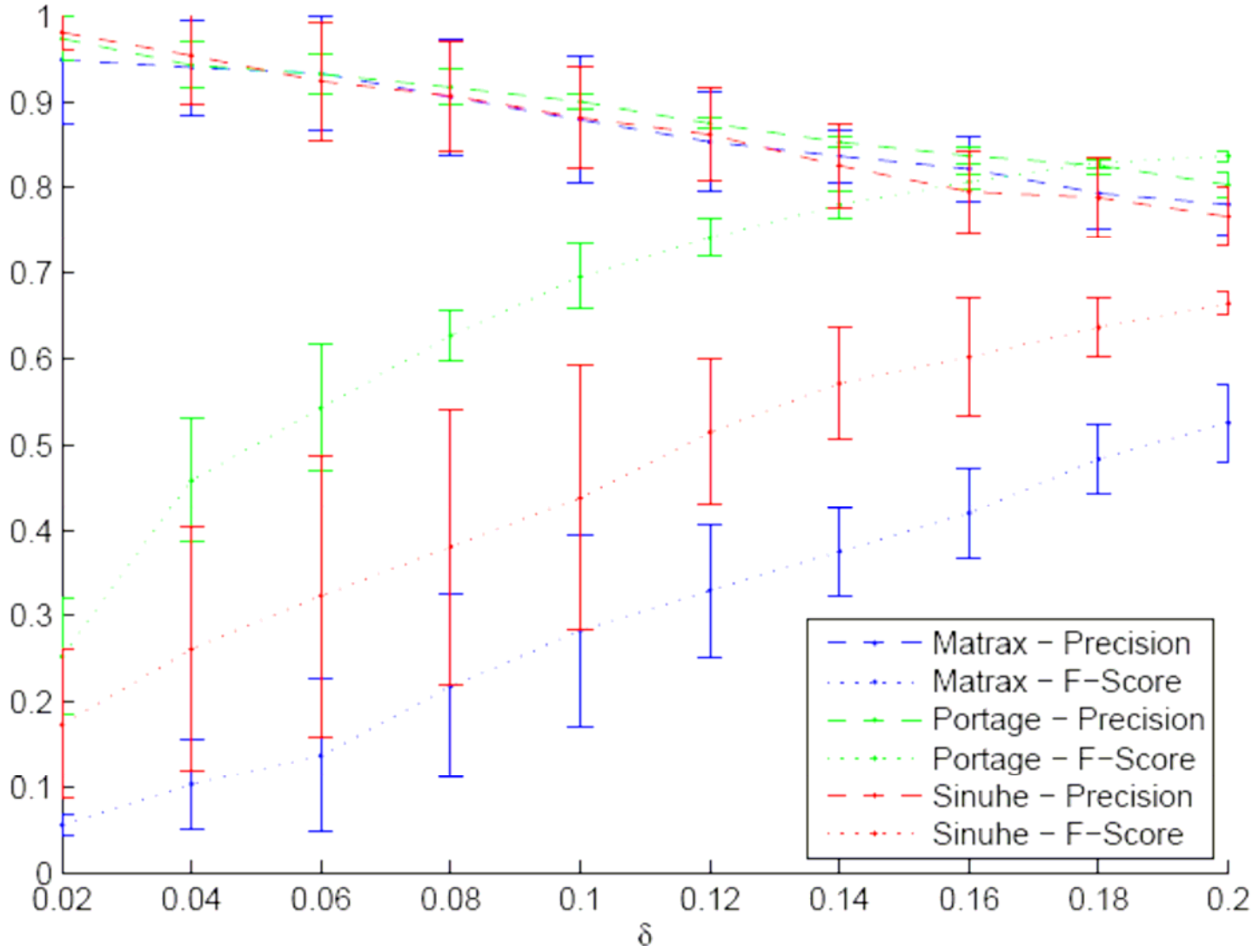
1. $L \leftarrow \min(\hat{\mathbf{y}}), U \leftarrow \max(\hat{\mathbf{y}});$
2. $S \leftarrow \{i \mid L \leq \hat{y}_{m+i} \leq U\};$
3. $\rho \leftarrow \text{median}(\hat{\mathbf{y}}_S), \hat{\delta} \leftarrow \dfrac{\left|\{i \mid \hat{y}_{m+i} \geq \rho, y_{m+i} \leq \tau\}\right|}{\left|\{j \mid \hat{y}_{m+j} \geq \rho\}\right|};$
4. if $\hat{\delta} = \delta$ or $L = U$ then return $\rho;$
5. else if $\hat{\delta} < \delta$ then $L \leftarrow \rho;$
6. else if $\hat{\delta} > \delta$ then $U \leftarrow \rho;$
7. goto 2;

Inductive Confidence Machines



xerox

Inductive Confidence Machines



xerox

Inductive Confidence Machines

- PLS x PLS+ICM (confidence levels = 95% and 90%)

τ	Model	Precision	Recall
2	PLS	92.86±0.81	91.84±0.81
	ICM $\delta = 0.05$	95.31±0.77	73.01±5.67
	ICM $\delta = 0.1$	91.37±0.97	99.43±1.13
3	PLS	91.45±1.81	25.01±3.54
	ICM $\delta = 0.05$	94.22±7.10	17.46±1.23
	ICM $\delta = 0.1$	88.18±5.97	30.68±1.27

- PLS+ICM x SVM multi-class and SVM binary (precision, confidence level = 90%)

τ	SVM 4-classes	SVM binary	PLS+ICM
2	91.60±0.62	90.99±0.95	90.88±1.26
3	50.09±1.61	69.19±0.97	87.94±7.33

Discussion

- **Results** considered to be **satisfactory**
  - Error would yield uncertainty in the boundaries between two adjacent categories in the 1-4 datasets
- **Results** for estimating a given type of score **are similar** across different systems and language pairs
- Results **correlate better** with human scores than those of metrics using reference translations
  - Also true for models trained in **different datasets**
- Using **ICM** to threshold good/bad translations - better than:
  - Using pre-defined threshold
  - Using classifiers to estimate 1-4 or good/bad scores

On-going work

- Further investigate uses for the most relevant features:

1. Most relevant features are not those usually considered in **SMT models**. We plan to investigate whether they could be useful to improve the translations quality, e.g.:
   - To **complement** existing features in SMT models.
   - To **rerank n-best lists** produced by SMT systems, which could make use of the features that are not local to single hypotheses.

On-going work

2. Automatic metrics such as NIST aim at simulating how humans evaluate translations. We plan to investigate our findings with **human annotation** for **MT evaluation**, e.g.:
   - To provide **additional features** to reference-based metrics based on ML algorithms, like (Albrecht, J. and R. Hwa, 2007)
   - To provide a **score** to be combined with other MT evaluation metrics, like ULC (Gimenez and Marquez, 2008)
   - To provide a **new evaluation metric** on itself, with some function to optimize the correlation with human annotations, without the need of reference translations.

Thanks!

Lucia Specia

lucia.specia@xrce.xerox.com

The source…

Researchers say swine flu has "full pandemic potential", spreading readily between people and is likely to go global in the next six to nine months, with one in three of the world's population infected.

BBC News, 12/05/09

Validity of the *P*-value

- Valid *p*-value: $P\{p(y)\leq\delta\} \leq \delta$
- Assume that the calibration data and an arbitrary new test example are i.i.d drawn according to a fixed distribution D
- The active calibration examples ($\hat{y}^*_{m+i} \geq \rho$) are drawn i.i.d from the conditional distribution $D^*:=D\{(x,y)|f(x) \geq \rho\}$
- Assume the current data sequence is produced by
  - Generating an unordered set
  - Assigning a permutation to it
- The probability of the test example being selected as the last one is $k^*!/(k^*+1)!=1/(k^*+1)$
- $p(y^*_{k+1}) \leq \delta$ if and only if $\alpha(\hat{y}^*_{k^*+1}, y_{k^*+1})$ is among the $\lfloor \delta(k^*+1) \rfloor$ largest αs

$$P\{p(y^*_{k^*+1}) \leq \delta\} = \frac{1}{k^*+1}\lfloor \delta(k^*+1) \rfloor \leq \delta$$