Confidence Estimation for Statistical Machine Translation

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur,

Cyril Goutte, Alex Kulesza, Alberto Sanchis, Nicola Ueffing

Confidence Estimation for MT

Try to determine whether MT output is correct or not, eg:

æßÇä ÑÝÖ ÇáÇÑ æÖÚ åíÆÉ	$\Rightarrow$	The weather is always perfect in Baltimore. $\uparrow$
Çáí ÇÓÝÑ Úä ÓÊÉ Êáì	$\Rightarrow$	the ninth of nine last January accusation $\downarrow$

Make judgements about individual translations

... not to be confused with confidence intervals or statistical significance

Motivation

- CE is useful for practical applications of imperfect NLP technologies - helpful to know when you are wrong, particularly when users are involved!
- extensive previous work in SR, eg spoken dialog systems
- almost no work outside SR...
- motivation for workshop: apply CE to another challenging area of NLP; assess performance and attempt to draw general conclusions

Two Problems with MT

Want to train and test CE methods on MT data labelled for correctness, but...

Problem 1: evaluation - difficult to automatically assign correct/incorrect labels; difficult and expensive to do so manually:

- set of correct translations is large and ill-defined; contrast with SR:
  - SR: January twelfth, 1999 / January 12, 1999
  - MT: John saw Mary / Mary was seen by John

  knowing one correct translation gives only weak clues about others

- all automatic evaluation measures exhibit high variability at the sentence level - in MT eval these are typically averaged over whole texts, but whole-text CE is not very interesting - typically want to work at sentence level or below

Two Problems with MT (cont)

Problem 2: MT output is bad - most translations are incorrect

Some examples:

- china from the world bank loans railway communications network
- the irish republican army (ira) in a report to the british and irish media issued a statement that the current situation and to promote northern ireland peace process , which made the decision and to make the decision had been informed by its subsidiary of the armed forces .
- he pointed out that the us proposal to lift the arms embargo on bosnia - herzegovina , ” which means that the ” international assistance can be brought to an end ” and ” major open conflict broke out again ” , which will be a heavy losses caused the serious consequences

Two Problems with MT (cont)

“Solutions”:

1. Evaluation: assess existing automatic error metrics and choose ones with highest correlation *at the sentence level* with human judgements, as measured on data collected in in-house evaluation exercise.

2. Low MT quality: redefine “correctness” as having an error level below a given threshold - distinguish between slightly bad and very bad translations.

   Justification: different error thresholds correspond to different potential applications, eg keywords for CLIR, rough semantics for gisting, etc.

CE for MT Setup

Learn the behaviour of the SMT group's base system (C $\longrightarrow$ E) on a corpus disjoint from its training set; test confidence estimates on a separate corpus

- raw examples of the form: (S, $\{$nbest hypotheses$\}$,$\{$ref trans$\}$)
- transform using automatic error metric and threshold τ into examples of the form: $(S, T_i, C_i), \;\; i = 1 \ldots n$, where:

$$C_i = \begin{cases} 1, & \text{error}(S, T_i, \{\text{ref trans}\}) \leq \tau \\ 0, & \text{else} \end{cases}$$

- perform experiments
- test on similarly transformed corpus

Experiments

For each of two levels of granularity - sentence and sub-sentence:

- Methods:
  - features
  - learning techniques
- Evaluations:
  - task-independent: strong CE versus weak CE
  - applications (sentence-level only)

Granularities

Sentence-level CE: learn from fuzzy match between T and $\{$ref trans$\}$. Applications:

- model combination $*$
- re-ranking $*$
- filtering for postediting
- active learning

Sub-sentence CE: learn from exact match between $w_i \in T$ and corresponding word in $\{$ref trans$\}$ under various definitions of “corresponding word” which parallel sentence-level error measures. Applications:

- highlighting for postediting
- hypothesis recombination: useful for SR, but much harder for MT due to reordering

Methods

Features - several classifications:

- dependent or not on base model
- dependent on S, T, or (S, T)
- knowledge source used

ML technique:

- none - use posterior probs from base model, or statistic over base model scores
- use a separate ML layer:
  - similar to stacking (Wolpert 1992)
  - modularity advantages over pure base model approach: ML layer can be retrained for different domains, or even reused for different systems; separate problem of picking best solution from that of determining its correctness
  - NB versus MLP

Task Independent Evaluation: Weak CE versus Strong CE

Weak CE - make binary correctness judgements only: $C(S,T)$

- in general need to tune performance of binary classifier $C(S,T)$ for particular application (and possibly even for each context) to minimize expected cost
- evaluation should reflect performance of $C(S,T)$ across different tuning thresholds t (not to be confused with translation error threshold τ!): use ROC curves (correct recall versus incorrect recall) and IROC (ROC integral)

Strong CE - estimate probabilities of correctness: $p(C=1|S,T)$

- broadly applicable to any application (any expected cost function) without requiring tuning - if probability estimates are accurate!
- evaluation:
  - indirect: discriminability over various thresholds t on $p(C=1|S,T)$
  - direct: accuracy of prob estimates on test corpus using cross entropy (NLL)

Application Evaluation

- Model combination: use correctness probabilities (strong CE) to combine outputs from baseline MT system and CMU (C $\longrightarrow$ E) MT system
- Reranking: use correctness probabilities or classifier tuning threshold to re-order hypotheses in nbest lists from base MT system.

Outline of Presentation

- Introduction (GF)
- Experimental Setup (CG)
- Sentence-level Experiments:
  - feature description (EF)
  - task-independent results (GF)
  - application results (SG)
- Sub-sentence Experiments (NU)
- MT Evaluation (AK)
- Conclusion (GF)

NB: no 1-1 correspondence between presenters and work presented!

Outline of Presentation

- Introduction (GF)
- Experimental Setup (CG)
- Sentence-level Experiments:
  - feature description (EF)
  - task-independent results (GF)
  - application results (SG)
- Sub-sentence Experiments (NU)
- MT Evaluation (AK)
- Conclusion (GF)

Experimental Setup

- Corpus Issues: Learning from 100Gb Data
- Machine Learning for Correctness Estimation
- Naive Bayes, Neural Networks and Maximum Entropy
- Bootstraping Error Bars in One Slide

Corpus

Output of two systems (ISI and CMU) trained for the Chinese-English task in the NIST MT evaluation 2003. Data split:

- Training: 993 sentences (NIST 2001 eval) — 4 refs
- Training (ISI): 4107 sentences from LDC corpus — 1 or 4 refs
- Development: 565 sentences from LDC corpus — 4 refs
- Test: 878 sentences (NIST 2002 eval) — 4 refs

For each source sentence, 101 to 16384 hypotheses (N-best) generated

Each proposed translation is one example to classify as correct or incorrect

100 N-best $\Rightarrow$ 510,000 training examples

1000 N-best $\Rightarrow$ 5,093,744 training examples

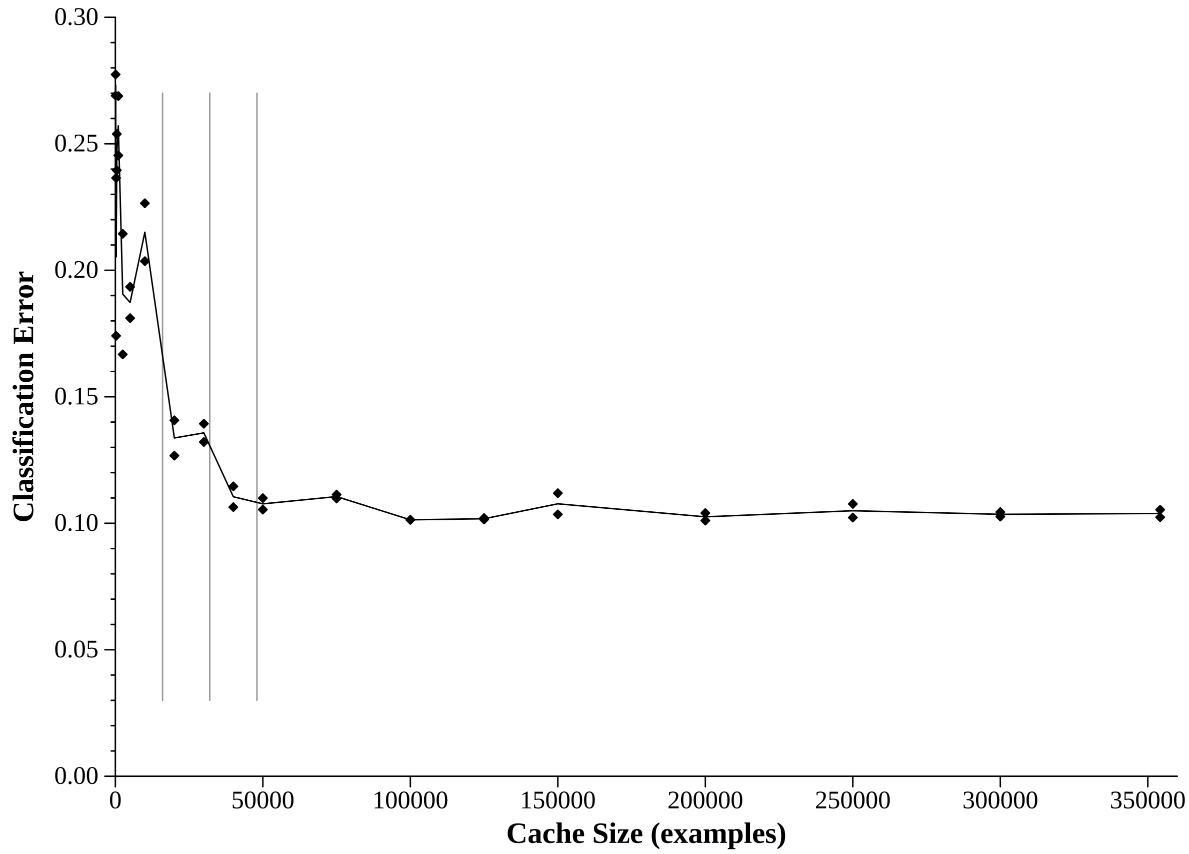
All N-best $\Rightarrow$ $\sim$80 million training examples

Learning from large size data

Dataset with more than $\sim$5 million examples will not fit in memory

$\Rightarrow$ Data caching, compression and parallel processing

1. keep data on disk, with small memory cache

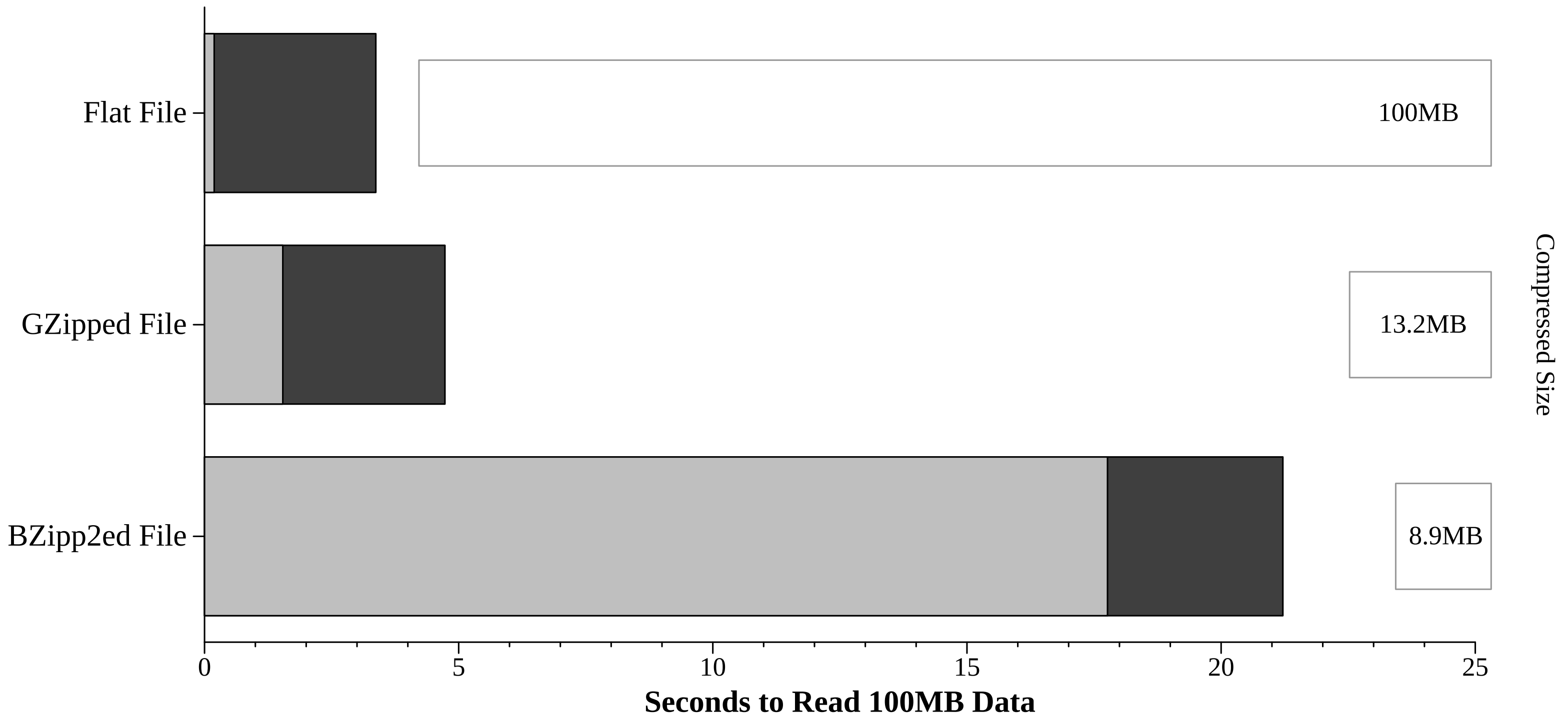


Learning from large size data

Dataset with more than ∼5 million examples will not fit in memory

⇒ Data caching, compression and parallel processing

1. keep data on disk, with small memory cache
2. gzip vs. bzip2 : 50% size loss, 5× to 10× speed improvement



Learning from large size data

Dataset with more than $\sim$5 million examples will not fit in memory

$\Rightarrow$ Data caching, compression and parallel processing

1. keep data on disk, with small memory cache
2. gzip vs. bzip2 : 50% size loss, $5\times$ to $10\times$ speed improvement
3. train several models in parallel to offset disk reads

$\longrightarrow$ Not all ML techniques may be practical

$\times$ Algorithms in $\mathcal{O}(N^3)$ complexity (SVM)

$\times$ Algorithms memorising large numbers of examples (kernelised perceptron)

$\surd$ Algorithms that run in $\mathcal{O}(N)$ time and $\mathcal{O}(1)$ space (RAM).

Machine Learning for (Conditional) Probability Estimation

We want to learn $P(c|\mathbf{x})$ from data

c: binary correctness indicator

$\mathbf{x}$: example hypothesis, represented by a number of features (to be defined)

The correctness of MT output is generally unknown for large data, but may be estimated using automatic scores

In our experiments, correctness is estimated by thresholding on:

WERg: Word error rate, normalised by the length of the Levenshtein alignment

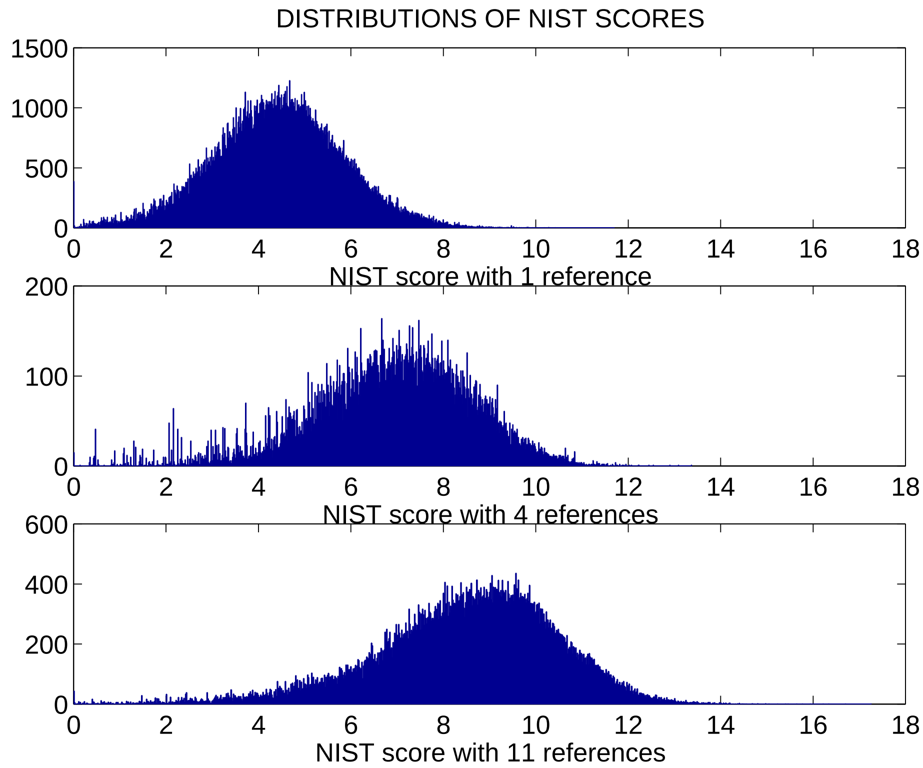
NIST: sentence-level NIST score (weighted average of n-gram precision)

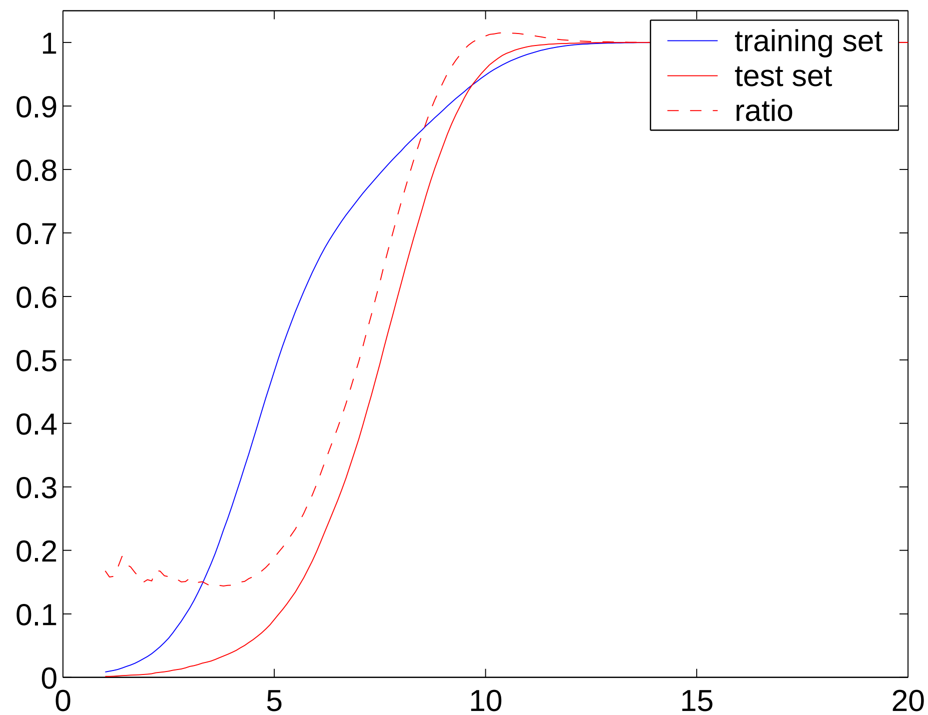
Do these measures agree with human correctness judgement? $\longrightarrow$ **Evaluation**

Correctness score and multiple references

We use the *sentence-based* score only

With more reference translations, scores automatically increase (and errors drop):





Scores of sentences with a single reference are scaled up to roughly match the distribution of scores of sentences with 4 reference translations

Naive Bayes

A generative model where features are assumed independent:

$$P(c|\mathbf{x}) \propto P(c)P(\mathbf{x}|c) \approx \widehat{P}(c) \prod_i \widehat{P}(x_i|c)$$

Continuous features x_i are discretised into $\sim$ 20 bins

Parameter estimation in two passes over the training set:

1. Calculate min, max and set number of bins and bin size for each feature
2. Estimate class-conditionals $\widehat{P}(x_i|c)$ by smoothing empirical frequencies

The smoothing algorithm is described by Sanchis, Juan and Vidal, Proc. ICASSP'03.

$\longrightarrow$ The importance of each individual feature x_i is assessed by evaluating the performance of a classifier using this feature alone: $\widehat{P}(c|x_i) \propto \widehat{P}(c)\widehat{P}(x_i|c)$

Multi Layer Perceptrons

A discriminative model generalising linear classifiers:

$$\widehat{P}(c|\mathbf{x}) = s\left(\mathbf{v}^c.h(\mathbf{W}.\mathbf{x})\right)$$

$\mathbf{W}$: *input layer* weights; $\mathbf{v}^c$: *output layer* weights; parameters $\theta = \{\mathbf{W}, \mathbf{v}^c\}$

$h(\cdot)$ non-linear transfer function; $s(\cdot)$ “softmax” layer (≈logistic regression)

Training by empirical loss minimisation using gradient descent

Gradient of continuous loss easily calculated using *back-propagation*

With large datasets, train using stochastic gradient descent — For each example $\left(\mathbf{x}^k, c^k\right)$, update parameters according to loss gradient:

$$\widehat{\theta} \leftarrow \widehat{\theta} - \eta \nabla_\theta \ell\left(\mathbf{x}^k, c^k\right)$$

The examples should be presented in random order.

May be quite fast for redundant data (but prone to local minima)

Multi Layer Perceptrons and Maximum Entropy

The following models involve log-linear combinations:

Maximum Entropy: $$P(c|\mathbf{x}) \propto \exp\left(\sum_m \lambda_m f_m(c,\mathbf{x})\right)$$

$f_m(c,\mathbf{x})$ various feature functions

Single Layer Perceptron + softmax: $$P(c|\mathbf{x}) \propto \exp\left(\sum_i w_i^c.x_i\right)$$

reduces to MaxEnt with $f_i(c,\mathbf{x})$ composed of x_i's and zeros.

Multi Layer Perceptron + softmax: $$P(c|\mathbf{x}) \propto \exp\left(\sum_j v_j^c.h\left(\mathbf{W}_{j.}.\mathbf{x}\right)\right)$$

reduces to MaxEnt if $\mathbf{W}$ fixed

can generalise MaxEnt to non-linear feature combinations

Bootstrapping Error Bars

From a true population distribution F, we seek a statistic $\theta = \phi(F)$
(eg $\theta = \mathrm{argmin}_\mu\, E(x-\mu)^2$)

We have a sample $\widehat{F}$, from which we estimate $\widehat{\theta} = \phi(\widehat{F})$
(eg $\widehat{\theta} = \mathrm{argmin}_\mu \sum (x^i - \mu)^2$)

How do we estimate the behaviour of $\widehat{\theta} - \theta$?

Bootstrap principle: replace F by $\widehat{F}$. (Efron, 1982; Efron&Tibshirani, 1993)
Sampling from $\widehat{F}$ = sampling (with replacement) from available data = “resampling”

For each “resample” F^*, get the corresponding statistic θ^*, and assume $(\theta^* - \widehat{\theta})$ behaves like $(\widehat{\theta} - \theta) \Rightarrow$ estimate bias, standard deviation, confidence interval, etc.

Error bars: find δ such that $P(|\theta^* - \widehat{\theta}| < \delta) = 1 - \alpha$ from the empirical distribution, then $[\widehat{\theta} - \delta, \widehat{\theta} + \delta]$ should have the required coverage.

Bootstrapping Error Bars

- From all the data $(y, c) \sim P(y, c)$, we may calculate the "true" performance E.
- From sample $S = \{y_i, c_i\}$, we may estimate this performance $\widehat{E} = f(S)$.
- *If we could* repeatedly sample from $P(y, c)$, we could obtain additional samples S, corresponding performance estimates, and finally obtain the distribution of $(\widehat{E} - E)$.

. . . but we can't.

- Instead we replace $P(y, c)$ by empirical $\widehat{P}(y, c) \propto \sum \delta(y_i, c_i)$.
- *Now we can* repeatedly sample some S^* from $\widehat{P}(y, c)$ and calculate E^*.

- Bootstrap principle: replace $(\widehat{E} - E)$ by $(E^* - \widehat{E})$
- For error bars: find $\Delta, P(|E^* - \widehat{E}| < \Delta) = 1 - a$ from many bootstrap replications, and use Δ as error bar for the estimate $\widehat{E}$

Outline of Presentation

- Introduction (GF)
- Experimental Setup (CG)
- Sentence-level Experiments:
  - feature description (EF)
  - task-independent results (GF)
  - application results (SG)
- Sub-sentence Experiments (NU)
- MT Evaluation (AK)
- Conclusion (GF)

Sentence-Level Experiments—Task-Independent Evaluation

- which features are best?
- how hard are different error measures/thresholds to learn?
  - NIST: 5% and 30% thresholds define correctness
  - WER: 5% and 30% thresholds define correctness
- which ML methods are best?
  - raw features vs NB vs MLP
  - regression vs classification
  - source-sentence-based normalization
  - learning curves

Tests

Corpus: NIST MT Eval, 993 source $\times$ 100, 1000 nbest lists:

N	num sent	NIST		WER	
		5%	30%	5%	30%
100	87,800	4.0%	34.4%	6.9%	34.1%
1000	876,831	3.2%	32.5%	5.7%	32.5%

Metrics:

- discriminability: ROC, AROC = $|IROC - .5| * 2$
- probability estimates: NLL

Single-Feature Comparison - Discriminability

P	NIST		WER	
5	44.92	**model1.1**	56.29	**BaseScore.0**
	38.14	searchfeat.4	54.56	searchfeat.2
	37.95	searchfeat.2	54.55	searchfeat.4
	37.83	searchfeat.3	53.05	searchfeat.3
	35.78	BaseScore.0	48.39	BaseFeatures.0
30	29.65	atal-ngram-0-6.3	34.61	BaseScore.0
	29.48	avg-nbestwordfeat.4	34.20	model1.1
	29.48	avg-nbestwordfeat.1	33.01	searchfeat.2
	29.36	avg-nbestwordfeat.5	32.76	searchfeat.4
	29.36	avg-nbestwordfeat.2	32.47	BaseFeatures.0

Single-Feature Comparison - Prob Estimates

P	NIST		WER	
5	0.2000	searchfeat.5	0.2761	searchfeat.4
	0.2004	nbestfeat.4	0.2765	BaseScore.1
	0.2005	searchfeat.4	0.2826	searchfeat.5
	0.2012	BaseScore.1	0.2829	searchfeat.3
	0.2016	BaseFeatures.1	0.2853	BaseFeatures.1
30	0.8574	nbestfeat.4	0.8453	BaseScore.1
	0.8721	atal-ngram-0-6.4	0.8515	searchfeat.4
	0.8776	avg-nbestwordfeat.2	0.8532	searchfeat.3
	0.8776	avg-nbestwordfeat.5	0.8541	searchfeat.5
	0.8777	avg-nbestwordfeat.3	0.8587	BaseFeatures.1

Single-Feature Comparison - Discriminability Attribution

B/N	S	T	S+T	ALL
B	22.07	16.61	13.77	15.86
N	15.96	7.85	25.25	11.55
ALL	18.09	9.92	14.91	13.63

S	T	S+T	ALL
14.14	11.43	11.73	12.10
11.93	20.65	14.32	17.46
12.70	18.47	12.00	14.87

B/N	S	T	S+T	ALL
B	29.69	36.71	23.04	27.05
N	16.84	26.00	28.41	23.23
ALL	21.31	28.54	23.58	25.08

S	T	S+T	ALL
19.85	19.92	13.32	15.86
11.49	10.96	19.06	11.65
14.40	13.08	13.90	13.68

Single-Feature Comparison - Prob Attribution

B/N	S	T	S+T	ALL
B	.2045	.2053	.2063	.2056
N	.2069	.2067	.2076	.2068
ALL	.2057	.2064	.2063	.2039

S	T	S+T	ALL
.9002	.9119	.9080	.9074
.9060	.8962	.9085	.8998
.9040	.8999	.9080	.8935

B/N	S	T	S+T	ALL
B	.3009	.2975	.3074	.3042
N	.3102	.3076	.3160	.3087
ALL	.3070	.3052	.3077	.3031

S	T	S+T	ALL
.8867	.8864	.9013	.8957
.9032	.9087	.9169	.9071
.8975	.9034	.9019	.8914

Error Measure Comparison

Average AROC over all (reasonable) MLP configurations, for $n = 1000$:

P	NIST	WER
5	53.79	52.60
30	48.69	41.69

ML Comparison

Features normalized over each nbest list versus non-normalized version:

P	NIST	WER
5	36.25	47.97
30	44.84	38.57

P	NIST	WER
5	53.79	52.60
30	48.69	41.69

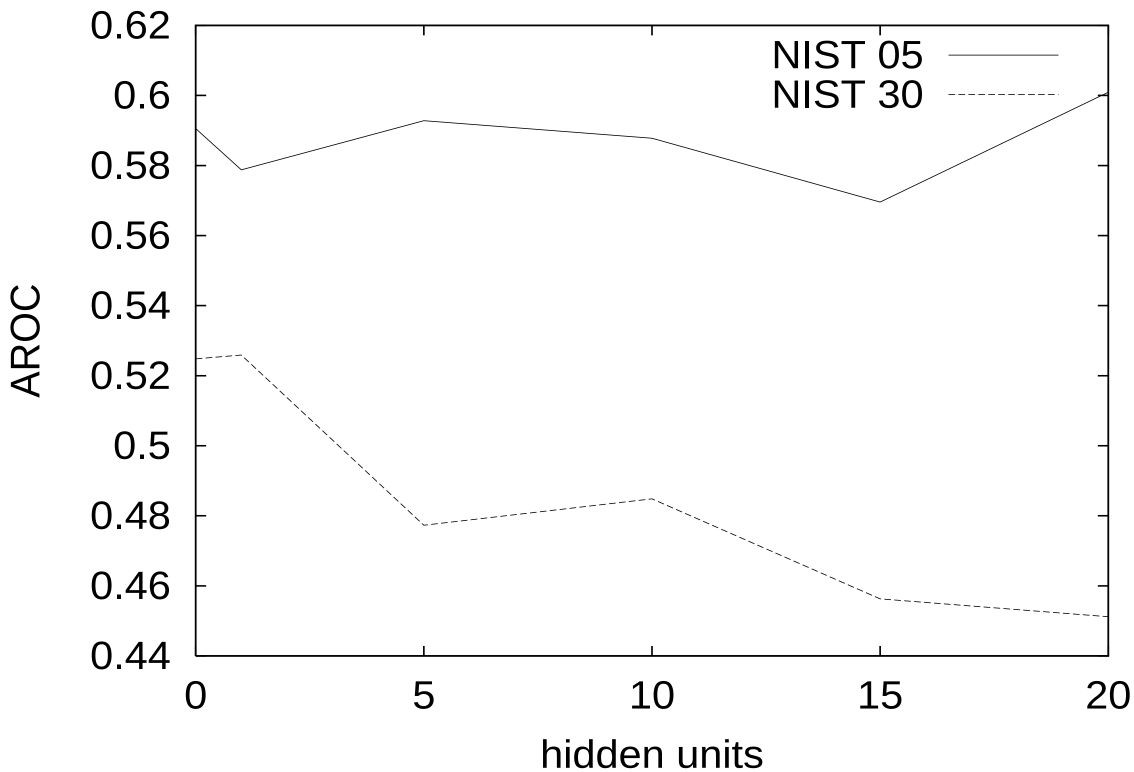
Regression versus classification:

P	NIST	WER
5	48.90	47.71
30	48.71	39.19

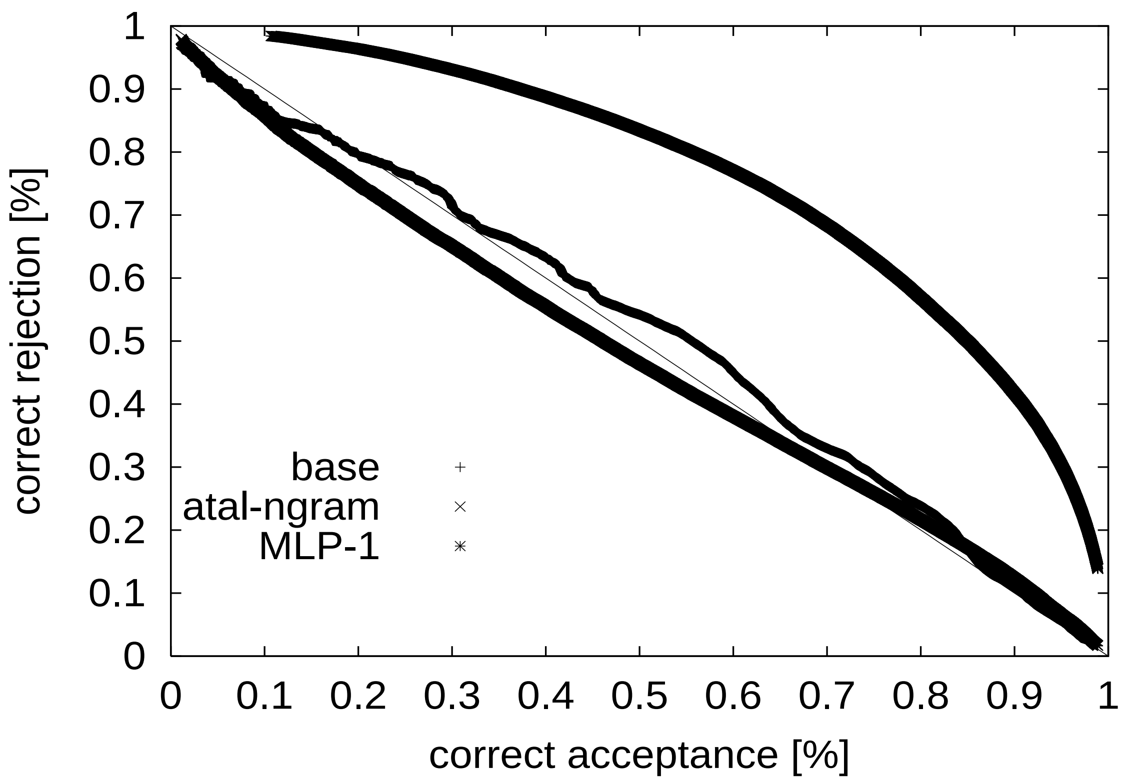
P	NIST	WER
5	58.76	62.49
30	48.76	44.20

ML Comparison: MLP Hidden Units

Features normalized over each nbest list versus non-normalized version:



ML Comparison: Raw feature vs MLP



Outline of Presentation

- Introduction (GF)
- Experimental Setup (CG)
- Sentence-level Experiments:
  - feature description (EF)
  - task-independent results (GF)
  - application results (SG)
- Sub-sentence Experiments (NU)
- MT Evaluation (AK)
- Conclusion (GF)

Applications for Sentence-level CE

- Re-ranking: ISI, CMU
- Model Combination: ISI + CMU

Challenges

- Sentence Level CE Goals: determine "goodness" of SMT translation hypothesis on a per sentence basis
- Difficulties:
  - evaluation: goodness = ?
  - re-ranking difficulty: CE model focusses on determining the probability of correctness of SMT results, not on ranking

Re-ranking: ISI

ISI	BLEU	NIST	aps-NIST	WERg
Baseline	30.81 (± .84)	9.29 (± .11)	7.47	0.619
CE-NIST	30.26 (± .90)	9.20 (± .12)	7.67	0.619
CE-WER	29.08 (± .85)	9.14 (± .12)	7.48	0.620
Oracle aps-NIST	30.36 (± .92)	9.21 (± .11)	9.51	0.538
Oracle WERg	30.36 (± .88)	9.21 (± .12)	8.56	0.465

Re-ranking: CMU

CMU	BLEU	NIST	aps-NIST	WERg
Baseline	17.39 (± .81)	7.50 (± .11)	6.89	0.700
CE-NIST	17.86 (± .76)	7.18 (± .11)	6.73	0.721
CE-WER	17.39 (± .78)	7.31 (± .12)	6.64	0.715
Oracle aps-NIST	22.96 (± .83)	8.59 (± .11)	8.55	0.675
Oracle WERg	21.17 (± .79)	7.86 (± .11)	7.52	0.608

Model Combination: CMU + ISI

Combination method: maximum score voting

ISI + CMU	BLEU	NIST	aps-NIST	WER-g
Baseline	30.81 ($\pm$ .84)	9.29 ($\pm$ .11)	7.47	0.619
Norm. base score	17.63 ($\pm$ .83)	7.53 ($\pm$ .11)	6.90	0.695
CE-NIST	22.31 ($\pm$ .99)	7.90 ($\pm$ .14)	7.36	0.684
CE-WER	28.37 ($\pm$ .91)	8.87 ($\pm$ .13)	7.14	0.641
Oracle aps-NIST	30.83 ($\pm$ .99)	9.52 ($\pm$ .11)	9.80	0.558
Oracle WERg	30.62 ($\pm$ .88)	9.21 ($\pm$ .12)	8.61	0.462

Sentence level CE – Conclusions

- Discriminability improvement: Yes
- Re-ranking: No
- Model combination: No
- Future challenges:
  - better sentence level SMT evaluation metrics
  - improve confidence features and ML approaches
  - more appropriate applications: filtering for postediting, active learning

Outline of Presentation

- Introduction (GF)
- Experimental Setup (CG)
- Sentence-level Experiments:
  - feature description (EF)
  - task-independent results (GF)
  - application results (SG)
- Sub-sentence Experiments (NU)
- MT Evaluation (AK)
- Conclusion (GF)

Overview

- Motivation
- Word Level Features
- Word Error Measures
- Experimental Results
- Outlook

Motivation

Sentence level confidence estimation:

- Sentence as a whole might be incorrect, but contain correct parts (only 30% of translations were rated 4 or 5 in our human evaluation exercise)
- Classification correct/incorrect easier on sub-sentence level than on sentence level
- Confidence estimates for sub-sentence level allow for recombination of different translation alternatives

Possible applications:

- Highlight incorrect words for post-editing
- Output only words with high confidence (e.g. in interactive translation environment)
- Recombination

Target Language Based Word Features		
Description	model dep.	blame
Identify incorrect parentheses and quotation marks	–	Erin
* Avg. of semantic similarity	+	John
* WordNet polysemy count	–	John
* WordNet polysemy count w.r.t. tagged corpus	–	John
* Relative frequency (in any target sentence position) (1)	–	Nicola
* Normalized rank sum (2)	–	Nicola
* Word posterior probability (3)	+	Nicola
* 1 – 3 for the exact target position	+/–	Nicola

Source/Target Language Based Word Features		
Description	model dep.	blame
Average Model1 Chinese-to-English log-probability over the entire source sentence	–	Erin

SMT Model Based Word Features	
Description	blame
* Relative frequency of word (aligned to the same source position(s))	Alberto
* Normalized rank sum (. . .)	Alberto
* Word posterior probability (. . .)	Alberto
Index of Alignment Template containing this word	John
Rule based or statistical translation (binary)	John

Semantic Features

- Average semantic similarity:
  - semantic similarity between words is the weighted sum of n-gram overlaps in WordNet glosses of the words and words related to them
  - compute average similarity between the word and the words aligned to same source word in the top 3 sentences
  - algorithm: Banerjee & Pedersen's [2002] adaptation to WordNet of Lesk's [1986] algorithm using conventional dictionaries
- WordNet polysemy count ($=$ number of senses stored in WordNet)
- WordNet polysemy count of senses occurring in tagged WordNet corpus

Word Posterior Probabilities and Related Measures I

Notation: target word e, target sentence e_1^I, source sentence f_1^J, alignment B_1^I

Word posterior probability: normalized sum of probabilities of all ’matching’ sentences in $\mathcal{S}(e,B)$:

$$\frac{1}{p(f_1^J)} \sum_{(e_1^I,\, B_1^I) \in \mathcal{S}(e,B)} p(e_1^I, B_1^I, f_1^J)$$

Relative frequency:

$$\frac{1}{N} \sum_{(e_1^I,\, B_1^I) \in \mathcal{S}(e,B)} 1$$

Rank sum:

$$\frac{2}{N(N+1)} \sum_{(e_1^I,\, B_1^I) \in \mathcal{S}(e,B)} (N + 1 - rank(e_1^I,\ B_1^I))$$

Word Posterior Probabilities and Related Measures II

Three (four) different variants of $\mathcal{S}(e, B)$:

- $\mathcal{S}(e,B) = \{(e_1^I,\ B_1^I) \mid e_i = e\}$

  word occurs in exactly this target position i

- $\mathcal{S}(e,B) = \{(e_1^I,\ B_1^I) \mid \exists\, i : (e_i, B_i) = (e, B)\}$

  word is aligned to source position(s) in B

- $\mathcal{S}(e,B) = \{(e_1^I,\ B_1^I) \mid \exists\, i : e_i = e\}$

  word occurs in the sentence

- (word occurs in a Levenshtein-aligned position)

Word Error Measures

Error Measure	word is correct iff ...
Pos	... it occurs in the reference in exactly this position
WER	... it is Levenshtein-aligned to itself in the reference
PER	... it occurs in the reference (bag of words)
Set	... it occurs in the reference (set of words)
n-gram	... this n-gram occurs in the reference ($n = 2, 3, 4$)

All measures except for n-gram exist in two variants:

1. Comparing to the pool of all references

2. Comparing to the nearest reference

Experimental Setup

Corpus Statistics (1000 best list)

	Source Sentences	Target Sentences	Running Words
Training	700	698 082	20 736 971
Develop	293	292 870	7 492 753
Test	878	876 831	26 360 766

Correct words [%] according to different error measures (pooled/nearest reference)

Error M.	Pos	WER	PER	Set	2-/3-/4-gram
Training	19.5 / 14.1	63.1 / 42.2	75.1 / 65.1	81.5 / 71.0	42.0 / 24.4 / 15.4
Develop	22.8 / 16.7	61.2 / 43.4	70.6 / 62.2	77.4 / 67.6	39.5 / 22.9 / 14.6
Test	21.7 / 15.5	62.3 / 42.5	73.6 / 63.8	80.7 / 70.0	41.5 / 24.4 / 15.5

Experimental Results – Single Features

Naive Bayes, Error Measure: PER

Feature		CER[%]	AROC[%]
Baseline		36.2	–
Any target position	WP / rank / rel.freq.	30.8-30.9	41.4-41.2
Model1		31.2	39.7
Aligned source position(s)	WP / rank / rel.freq.	31.9	39.0-38.8
Fixed target position	WP / rank / rel.freq.	32.5-32.7	37.7-37.2
AT identity		33.1	34.5
All		29.6	47.2

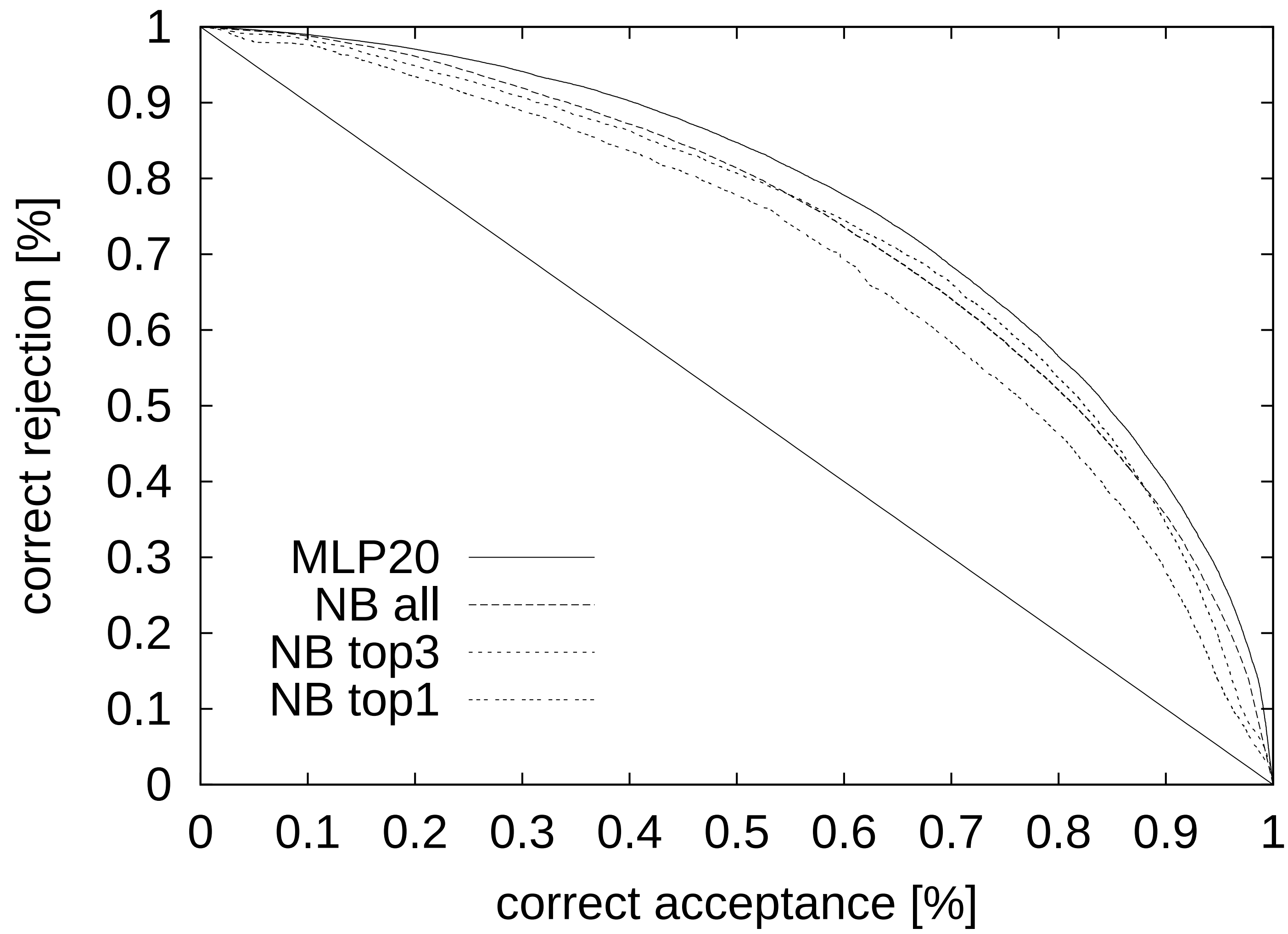
Experimental Results – AROC

AROC values [%] for different error measures

ML	features	WER	PER	Set
Naive Bayes	WP any + WP align + Model1	37.0	46.6	60.8
	all	38.2	47.2	61.4
MLP 20 hu	all	40.6	53.1	65.7

Experimental Results – ROC

ROC for PER



Recombination

Idea:

- Search criterion based on confidence estimation for words

  $\Rightarrow$ recombination of different translation hypotheses

Problems:

1. Sentence length

2. Selection criterion for target words:

   – best word in each target position: might cause inconsistencies, because same word can be selected twice

   – best target word for each source word: word order?

Possible solutions:

1. Normalization by sentence length

2. Represent search space by word graph and determine best path

Outlook

- Try more features (sentence level confidence estimate, target language n-gram probabilities, word identity, Levenshtein alignment to best hypothesis/center hypothesis, word posteriors according to Levenshtein alignment, ...)
- Recombination of hypotheses using confidence estimation

Outline of Presentation

- Introduction (GF)
- Experimental Setup (CG)
- Sentence-level Experiments:
  - feature description (EF)
  - task-independent results (GF)
  - application results (SG)
- Sub-sentence Experiments (NU)
- MT Evaluation (AK)
- Conclusion (GF)

Sentence-Level MT Evaluation

- Any large scale learning task must, to be reasonable, rely on an automatic evaluation metric of some kind:
  - Humans are slow
  - and expensive.
- Many metrics have been proposed – NIST, BLEU, etc. – but have been typically evaluated on a document or corpus level
- Our task requires accurate, automatic, =bf sentence-level evaluations.
- How to choose (or design) such a metric?
  - Score should reflect level of adequacy for particular applications
  - Estimated by correlation with **human** judgements of task adequacy

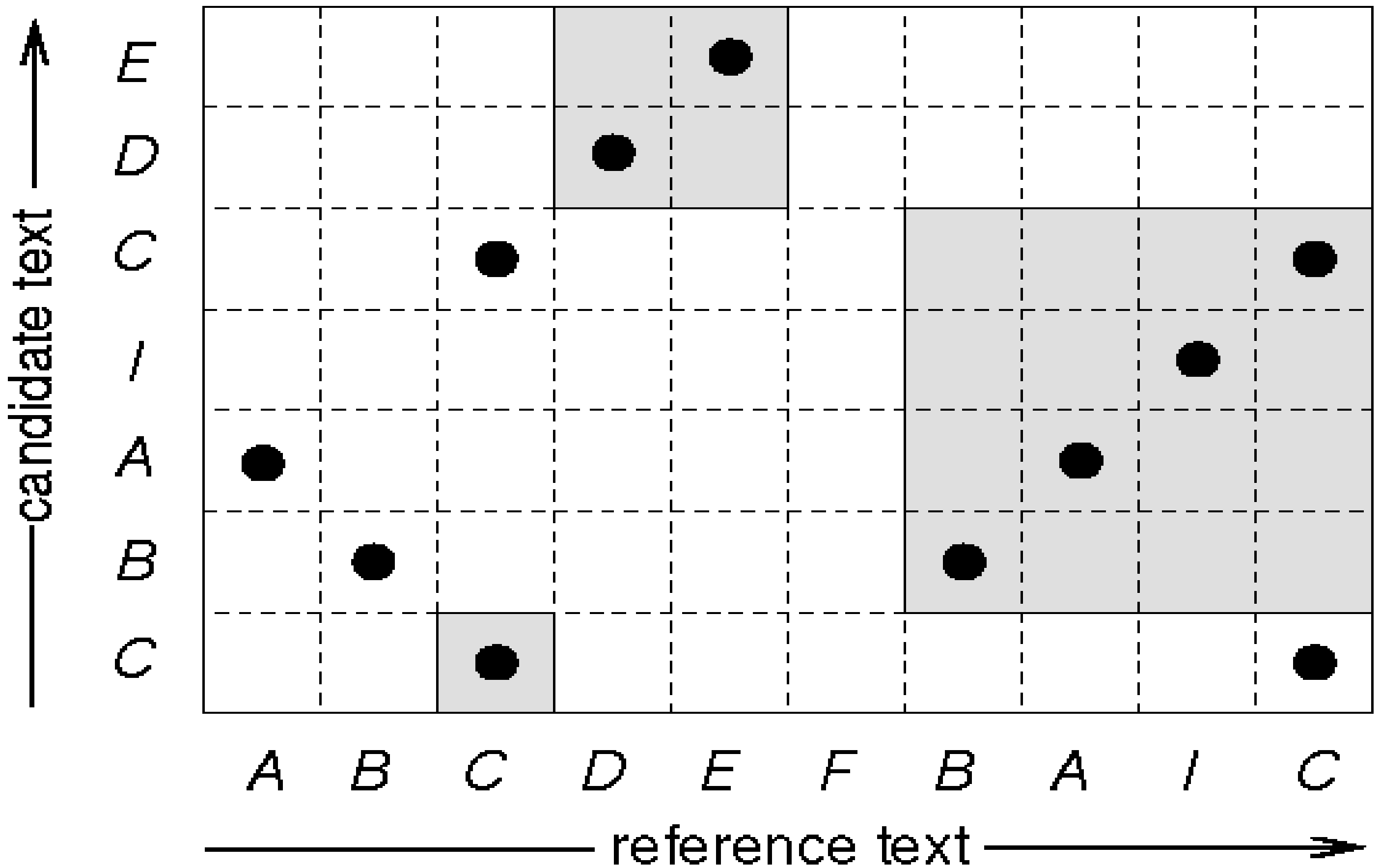
Automatic Error Metrics

- An error metric maps a hypothesis translation and a set of reference translations to a score – for our task a “translation” is one sentence.
- Metrics considered:
  - WER: Word error rate, computed as the minimum number of insertions, deletions, and substitutions required to transform the hypothesis into any reference (Levenshtein/edit distance), normalized by reference length.
  - WER-g: As above, but normalized by the total length of the alignment (insertions, deletions, substitutions, and matches).
  - PER: Position-independent error rate; treats both hypotheses and references as unordered bags of words and counts the necessary operations to make them equal. Normalized by reference length.

Automatic Error Metrics

- More metrics:
  - BLEU: The geometric mean of hypothesis n-gram precision for $1 \leq n \leq 4$, multiplied by an exponentially decaying length penalty, to compensate for short, high-precision translations ("the").
    - Smoothed precisions
    - Adjusted length penalty
  - NIST: The **arithmetic** mean of hypothesis n-gram precisions, weighted by n-gram frequencies in a fixed corpus (effectively, less common n-grams receive greater emphasis). Also uses a length penalty.
  - F-Measure: The harmonic mean of precision and recall, where the size of the match between hypothesis and reference is the maximum of $\sqrt[k]{\sum |r_i|^k}$ over all sets $M = \{r_1, ..., r_n\}$ of non-conflicting matched runs of words. $(k = 1)$

Automatic Error Metrics



Human Evaluation Protocol

- Human evaluations collected via a live server/client system.
- The system distributes sentences so as to maximize the number of sentences receiving scores from two users.
- Designed to optimize the process on both ends:
  - Users can evaluate as much or as little as they like, at any time.
  - Evaluation data is immediately accessible for analysis.

Human Evaluation Protocol

```
************************************************************************
    Human MT Eval Client
************************************************************************

  Hypothesis:

    ( washington ) , comprehensive report the latest issue of the new
    yorker " weekly , iraq ’s intelligence agencies responsible for
    many years and 911 incident osama bin laden under the leadership of
    the al qaeda maintain close ties .

  Reference:

    comprehensive report , washington -- the latest issue of new yorker
    magazine suggests that iraqi intelligence has been in close touch
    with top officials in al @-@ qaida group for years . the al @-@
    qaida group is believed to have masterminded the 911 incident .

  Enter your rating (1-5), ’h’ for help, or ’q’ to quit:
```

Human Evaluation Protocol

- Evaluation scale (1-5) is described as follows:

  Reference ex: *bob walked the dog.*

  - 1: Useless; captures absolutely none of the reference's meaning.

    ex: *franklin is a doctor.* **Satisfies no task.**

  - 2: Poor; contains a few key words, but little or no meaning.

    ex: *dog banana walk.* **"Bag of words" – IR, etc.**

  - 3: Mediocre; contains some meaning, but with serious errors.

    ex: *the dog walked bob.* **Gisting**

  - 4: Acceptable; captures most of the meaning with only small errors.

    ex: *bob walk the dog.* **Human post-processing**

  - 5: Human quality; captures all of the reference's meaning.

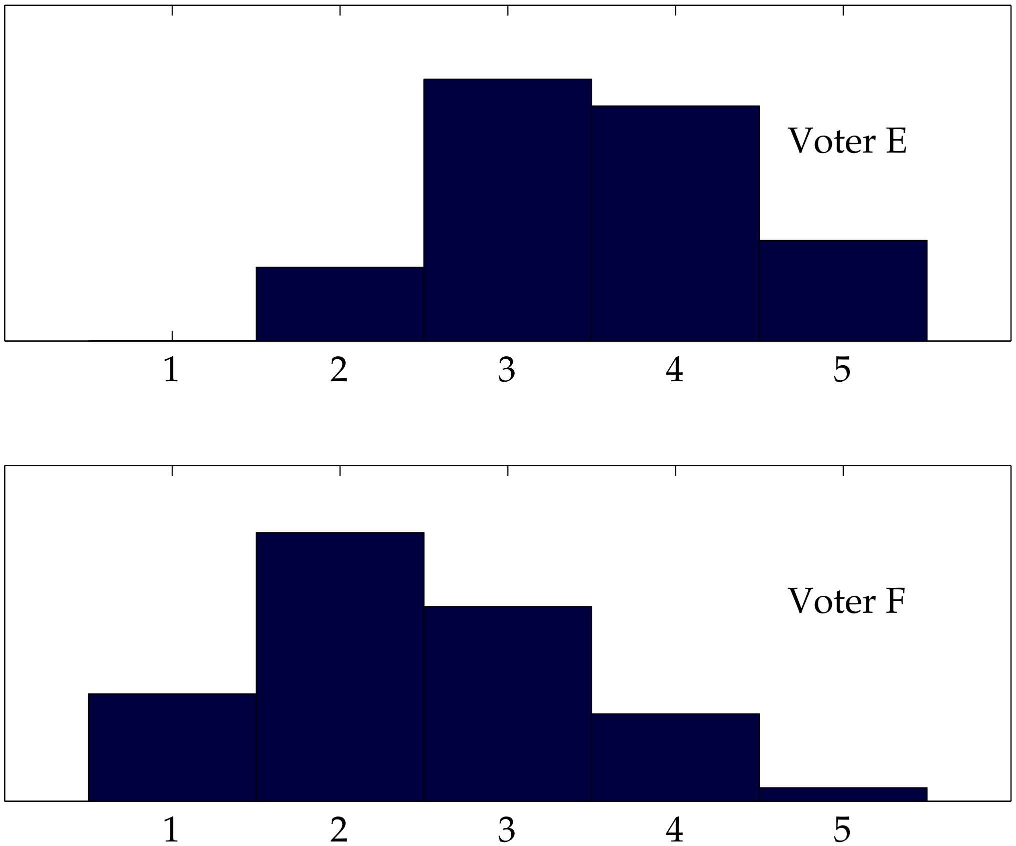
    ex: *bob took the dog for a walk.* **General use**

Data Collection Results

- 29 users
- Approximately 20 user-hours logged
- 705 sentences scored, each by two users
  - 72 calibration sentences
  - **633 hypotheses scored**
- Scoring rate of 74 sentences/hour suggests feasibility of larger-scale human evaluation data collection.

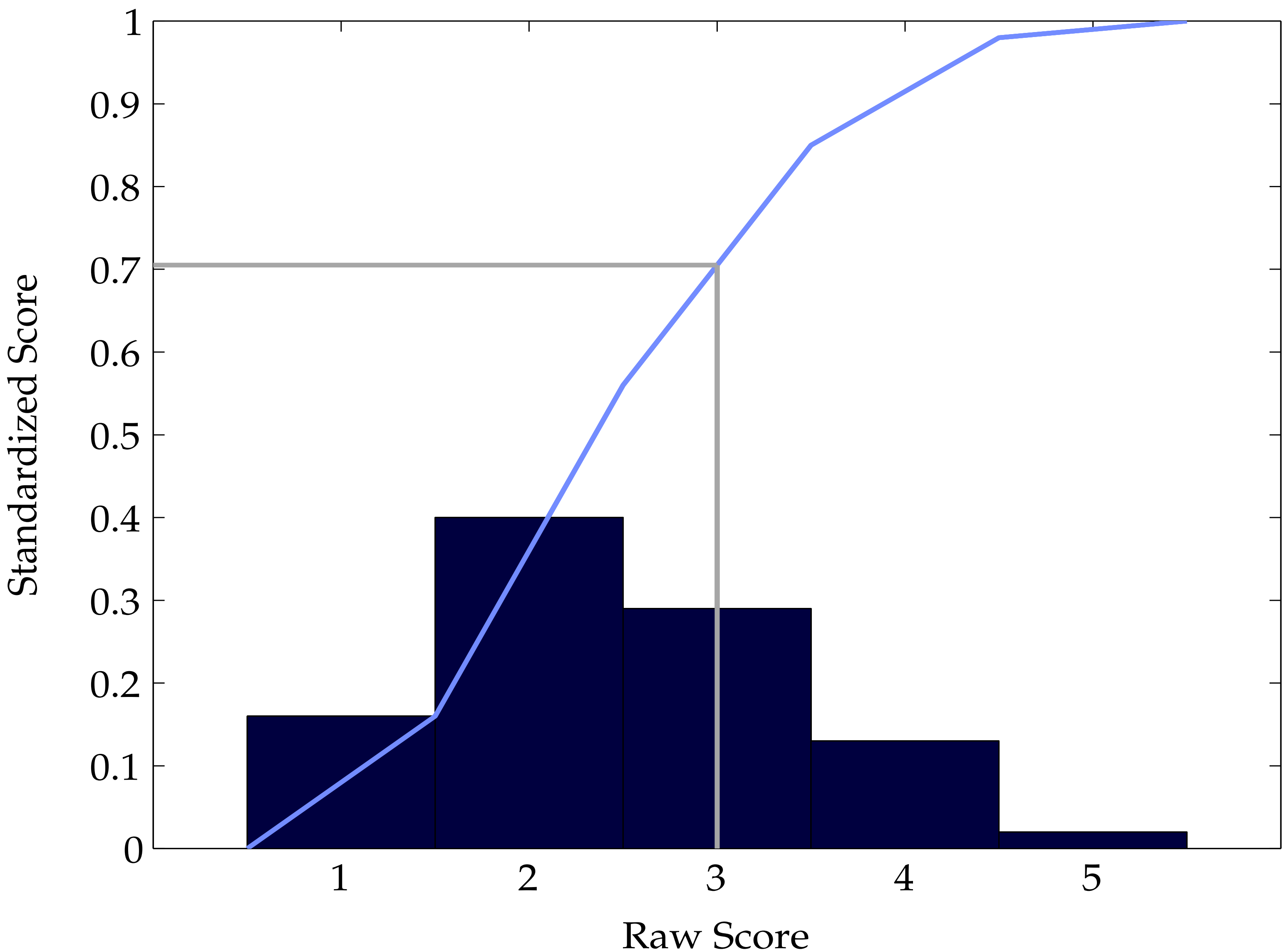
Score Standardization

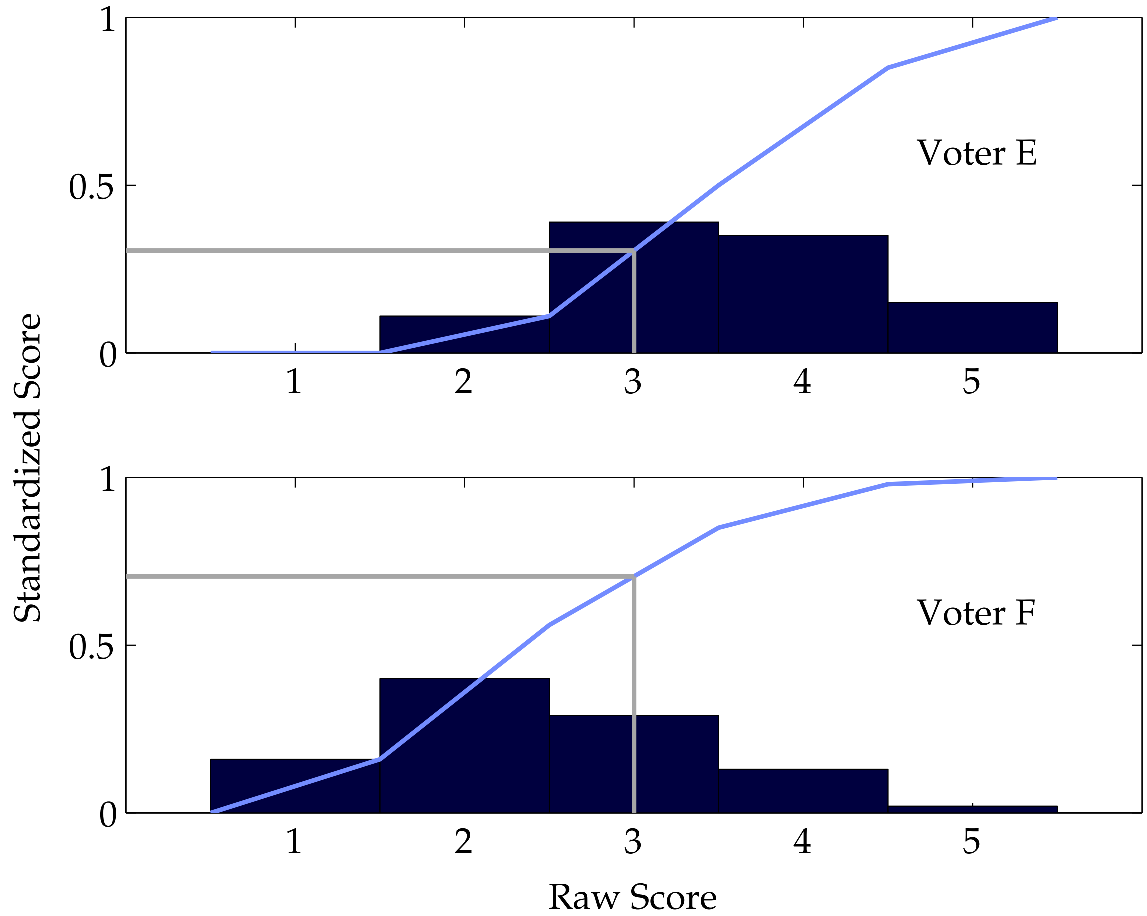
- Despite guidelines, voters differ:



- To compensate, raw scores are converted to approximate percentiles. (Eisner)

Score Standardization





Score Standardization

- When generating an “average” human score, the percentiles are weighted with the total number of hypotheses scored by each user.

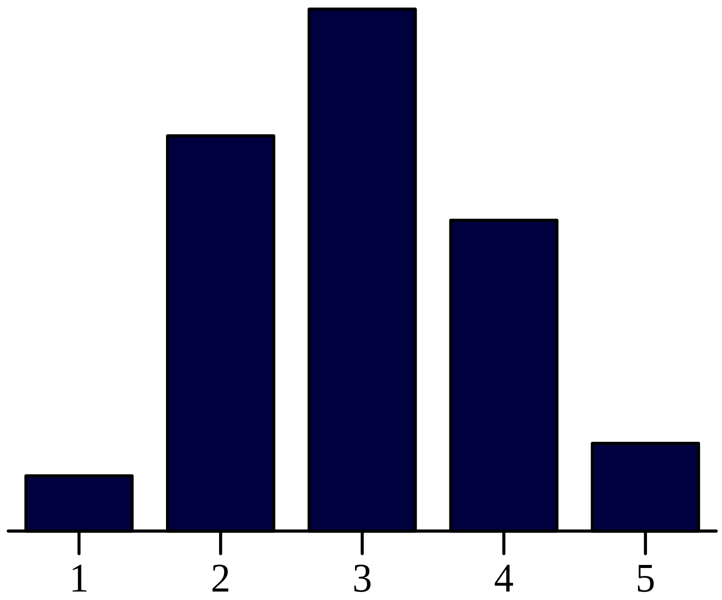
Summary of Human Data

- Standardization increases inter-annotator correlation from 0.433 to 0.463

Confusion:

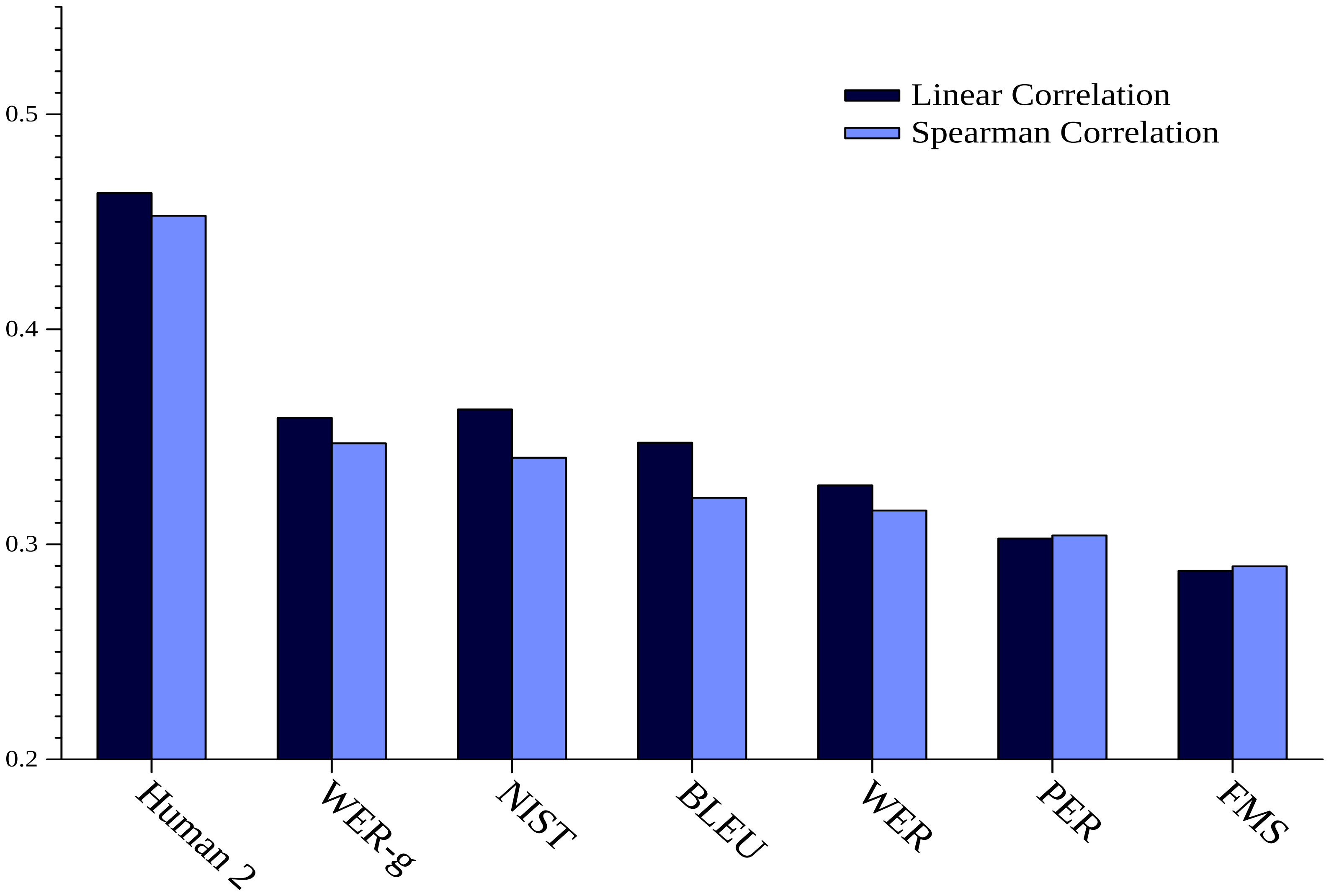
	1	2	3	4	5
1	2	14	9	1	
2	16	73	67	17	2
3	6	72	110	46	12
4	1	27	41	61	16
5		4	9	16	11

Overall distribution:

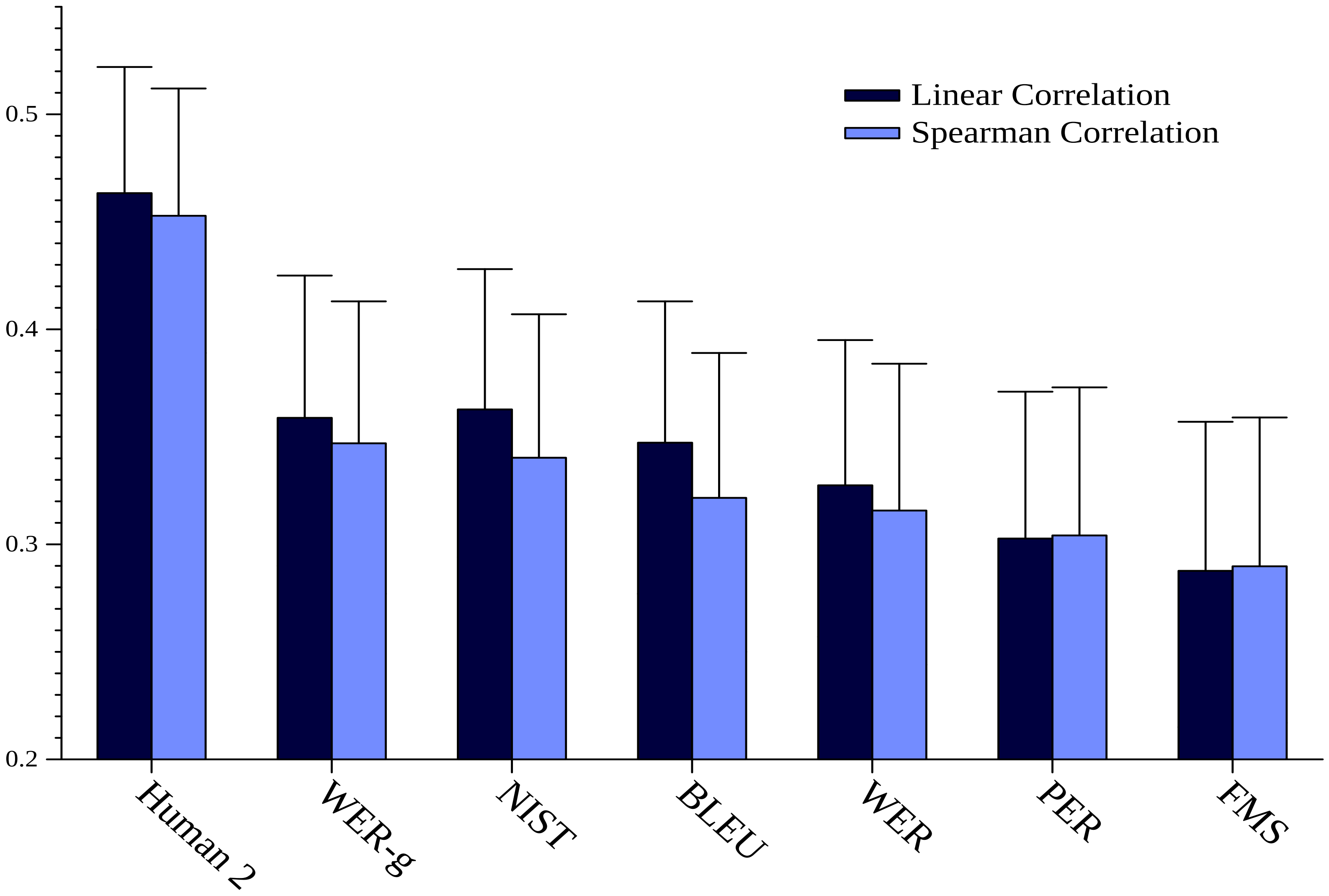


(bins in proportion to raw score distribution)

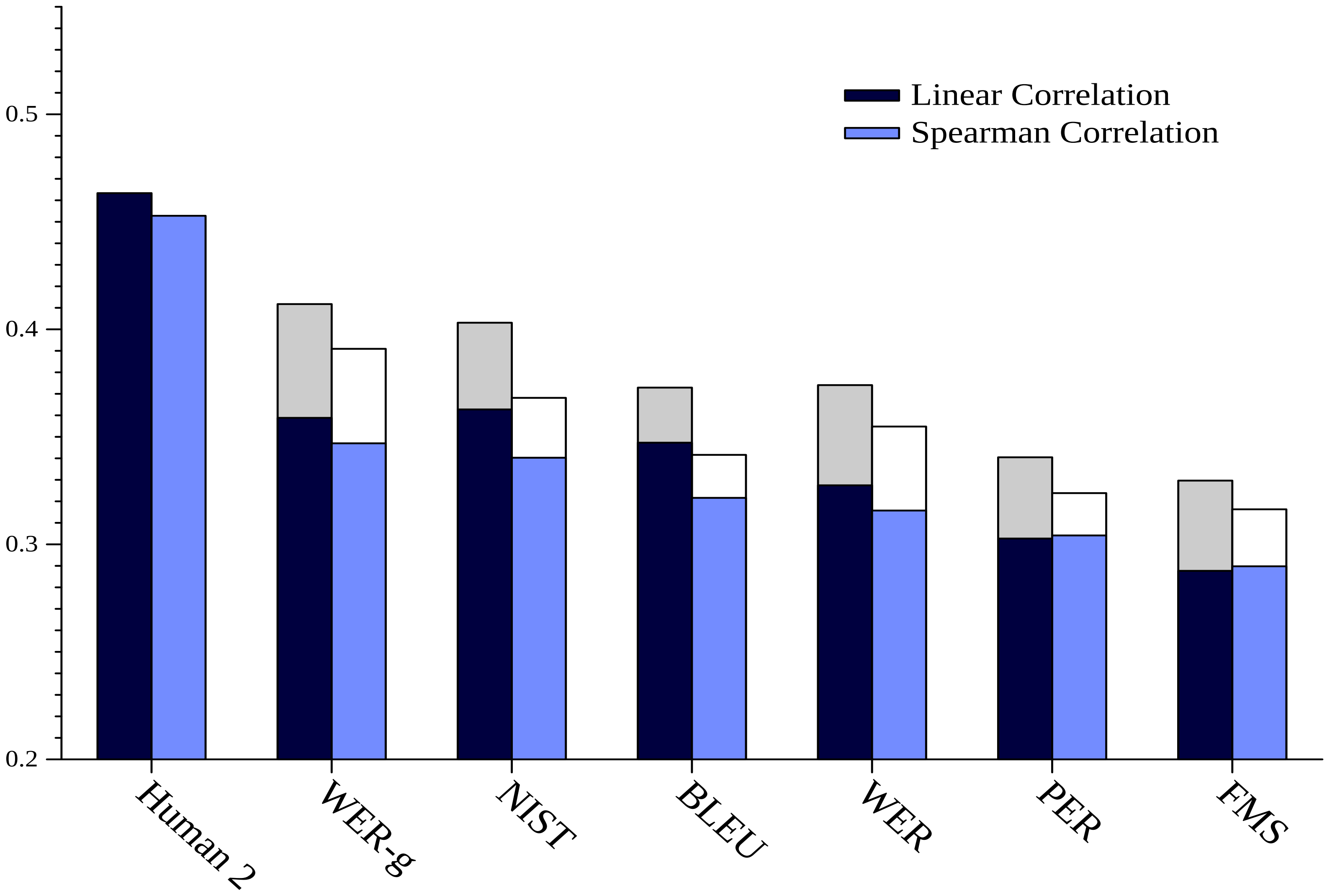
Automatic Measure Results



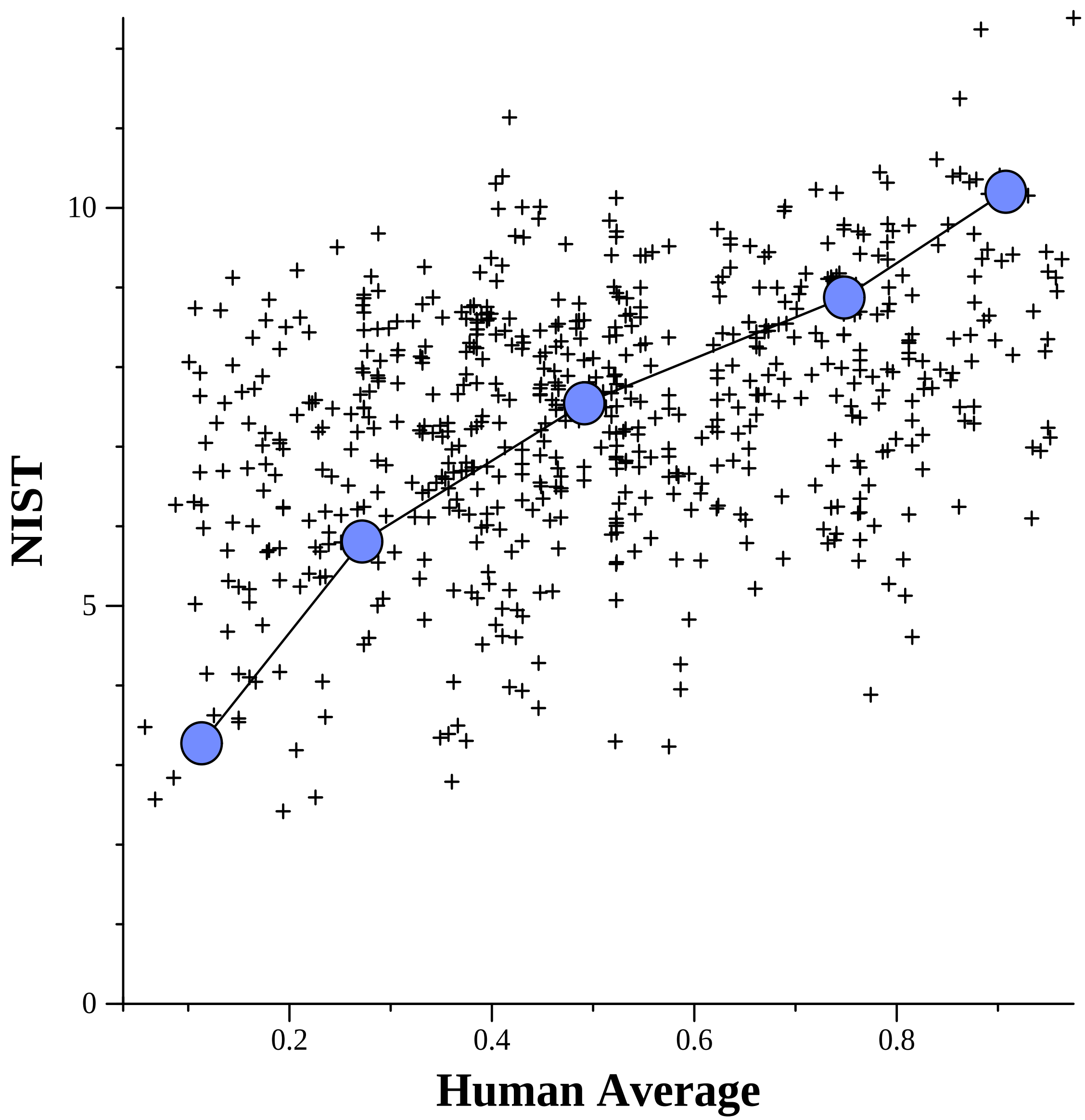
Automatic Measure Results



Automatic Measure Results



Automatic Measure Results



Automatic Measure Results

- Further data collection:
  1. More sentences (shrink error bars)
  2. More votes per sentence (reduce noise, increase correlation)
- Better metrics **at the sentence level**

Outline of Presentation

- Introduction (GF)
- Experimental Setup (CG)
- Sentence-level Experiments:
  - feature description (EF)
  - task-independent results (GF)
  - application results (SG)
- Sub-sentence Experiments (NU)
- MT Evaluation (AK)
- Conclusion (GF)

Summary of Results

Sentence-level CE:

- adding ML layer significantly improves discriminability over baseline approach
- no significant improvement on applications tried (model combination and re-ranking)

Sub-sentence CE:

- ML layer significantly improves discriminability over baseline approach - more improvement with rich feature sets and more hidden units

Human evaluation:

- no significant difference between error measures on our dataset
- inter-annotator agreement low but distinguishable from auto measures

Status and Future Work

Cannot claim that CE for MT is useful yet. Need better solutions to two basic problems:

- better evaluation metrics at the sentence level (or massive amounts of human annotation)
- better MT output would make the problem more clearly defined

Future directions:

- try a filtering application
- sub-sentence CE for recombination