Efficient Elicitation of Annotations for Human Evaluation of Machine Translation

Keisuke Sakaguchi*, **Matt Post**†, **Benjamin Van Durme**†
* Center for Language and Speech Processing
† Human Language Technology Center of Excellence
Johns Hopkins University, Baltimore, Maryland
{keisuke,post,vandurme}@cs.jhu.edu

Abstract

A main output of the annual Workshop on Statistical Machine Translation (WMT) is a ranking of the systems that participated in its shared translation tasks, produced by aggregating pairwise sentence-level comparisons collected from human judges. Over the past few years, there have been a number of tweaks to the aggregation formula in attempts to address issues arising from the inherent ambiguity and subjectivity of the task, as well as weaknesses in the proposed models and the manner of model selection.

We continue this line of work by adapting the TrueSkill™ algorithm — an online approach for modeling the relative skills of players in ongoing competitions, such as Microsoft's Xbox Live — to the human evaluation of machine translation output. Our experimental results show that TrueSkill outperforms other recently proposed models on accuracy, and also can significantly reduce the number of pairwise annotations that need to be collected by sampling non-uniformly from the space of system competitions.

1 Introduction

The Workshop on Statistical Machine Translation (WMT) has long been a central event in the machine translation (MT) community for the evaluation of MT output. It hosts an annual set of shared translation tasks focused mostly on the translation of western European languages. One of its main functions is to publish a ranking of the systems for each task, which are produced by aggregating a large number of human judgments of sentence-level pairwise rankings of system outputs. While the performance on many automatic metrics is also reported (e.g., BLEU (Papineni et al., 2002)), the human evaluation is considered primary, and is in fact used as the gold standard for its metrics task, where evaluation metrics are evaluated.

#	score	range	system
1	0.638	1	UEDIN-HEAFIELD
2	0.604	2-3	UEDIN
	0.591	2-3	ONLINE-B
4	0.571	4-5	LIMSI-SOUL
	0.562	4-5	KIT
	0.541	5-6	ONLINE-A
7	0.512	7	MES-SIMPLIFIED
8	0.486	8	DCU
9	0.439	9-10	RWTH
	0.429	9-11	CMU-T2T
	0.420	10-11	CU-ZEMAN
12	0.389	12	JHU
13	0.322	13	SHEF-WPROA

Table 1: System rankings presented as clusters (WMT13 French-English competition). The *score* column is the percentage of time each system was judged better across its comparisons (§2.1).

In machine translation, the longstanding disagreements about evaluation measures do not go away when moving from automatic metrics to human judges. This is due in no small part to the inherent ambiguity and subjectivity of the task, but also arises from the particular way that the WMT organizers produce the rankings. The system-level rankings are produced by collecting pairwise sentence-level comparisons between system outputs. These are then aggregated to produce a complete ordering of all systems, or, more recently, a partial ordering (Koehn, 2012), with systems clustered where they cannot be distinguished in a statistically significant way (Table 1, taken from Bojar et al. (2013)).

A number of problems have been noted with this approach. The first has to do with the nature of ranking itself. Over the past few years, the WMT organizers have introduced a number of minor tweaks to the ranking algorithm (§2) in reaction to largely intuitive arguments that have been

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 1–11,
Baltimore, Maryland USA, June 26–27, 2014.

raised about how the evaluation is conducted (Bojar et al., 2011; Lopez, 2012). While these tweaks have been sensible (and later corroborated), Hopkins and May (2013) point out that this is essentially a model selection task, and should properly be driven by empirical performance on held-out data according to some metric. Instead of intuition, they suggest perplexity, and show that a novel graphical model outperforms existing approaches on that metric, with less amount of data.

A second problem is the deficiency of the models used to produce the ranking, which work by computing simple ratios of wins (and, optionally, ties) to losses. Such approaches do not consider the relative difficulty of system matchups, and thus leave open the possibility that a system is ranked highly from the luck of comparisons against poorer opponents.

Third, a large number of judgments need to be collected in order to separate the systems into clusters to produce a partial ranking. The sheer size of the space of possible comparisons (all pairs of systems times the number of segments in the test set) requires sampling from this space and distributing the annotations across a number of judges. Even still, the number of judgments needed to produce statistically significant rankings like those in Table 1 grows quadratically in the number of participating systems (Koehn, 2012), often forcing the use of paid, lower-quality annotators hired on Amazon's Mechanical Turk. Part of the problem is that the sampling strategy collects data uniformly across system pairings. Intuitively, we should need many fewer annotations between systems with divergent base performance levels, instead focusing the collection effort on system pairs whose performance is more matched, in order to tease out the gaps between similarly-performing systems. Why spend precious human time on redundantly affirming predictable outcomes?

To address these issues, we developed a variation of the TrueSkill model (Herbrich et al., 2006), an adaptative model of competitions originally developed for the Xbox Live online gaming community. It assumes that each player's skill level follows a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, in which μ represents a player's mean performance, and σ^2 the system's uncertainty about its current estimate of this mean. These values are updated after each "game" (in our case, the value of a ternary judgment) in proportion to how surprising the outcome is. TrueSkill has been adapted to a number of areas, including chess, advertising, and academic conference management.

The rest of this paper provides an empirical comparison of a number of models of human evaluation (§2). We evaluate on perplexity and also on accuracy, showing that the two are not always correlated, and arguing for the primacy of the latter (§3). We find that TrueSkill outperforms other models (§4). Moreover, TrueSkill also allows us to drastically reduce the amount of data that needs to be collected by sampling non-uniformly from the space of all competitions (§5), which also allows for greater separation of the systems into ranked clusters (§6).

2 Models

Before introducing our adaptation of the TrueSkill model for ranking translation systems with human judgments (§2.3), we describe two comparisons: the "Expected Wins" model used in recent evaluations, and the Bayesian model proposed by Hopkins and May (§2.2).

As we described briefly in the introduction, WMT produces system rankings by aggregating sentence-level ternary judgments of the form:

$$(i, S_1, S_2, \pi)$$

where i is the source segment (id), S_1 and S_2 are the system pair drawn from a set of systems $\{S\}$, and $\pi \in \{<, >, =\}$ denotes whether the first system was judged to be better than, worse than, or equivalent to the second. These ternary judgments are obtained by presenting judges with a randomly-selected input sentence and the reference, followed by *five* randomly-selected translations of that sentence. Annotators are asked to rank these systems from best (rank 1) to worst (rank 5), ties permitted, and with no meaning ascribed to the absolute values or differences between ranks. This is done to accelerate data collection, since it yields ten pairwise comparisons per ranking. Tens of thousands of judgments of this form constitute the raw data used to compute the system-level rankings. All the work described in this section is computed over these pairwise comparisons, which are treated as if they were collected independently.

2.1 Expected Wins

The "Expected Wins" model computes the percentage of times that each system wins in its

pairwise comparisons. Let A be the complete set of annotations or judgments of the form $\{i, S_1, S_2, \pi_R\}$. We assume these judgments have been converted into a normal form where S_1 is either the winner or is tied with S_2, and therefore $\pi_R \in \{<, =\}$. Let $\delta(x, y)$ be the Kronecker delta function.[1] We then define the function:

$$\text{wins}(S_i, S_j) = \sum_{n=1}^{|A|} \delta(S_i, S_1^{(n)})\delta(S_j, S_2^{(n)})\delta(\pi_R^{(n)}, <)$$

which counts the number of annotations for which system S_i was ranked better than system S_j. We define a single-variable version that marginalizes over all annotations:

$$\text{wins}(S_i) = \sum_{S_j \neq S_i} \text{wins}(S_i, S_j)$$

We also define analogous functions for *loses* and *ties*. Until the WMT11 evaluation (Callison-Burch et al., 2011), the score for each system S_i was computed as follows:

$$\text{score}(S_i) = \frac{\text{wins}(S_i) + \text{ties}(S_i)}{\text{wins}(S_i) + \text{ties}(S_i) + \text{loses}(S_i)}$$

Bojar et al. (2011) suggested that the inclusion of ties biased the results, due to their large numbers, the underlying similarity of many of the models, and the fact that they are counted for both systems in the tie, and proposed the following modified scoring function:

$$\text{score}(S_i) = \frac{1}{|\{S\}|} \sum_{S_j \neq S_i} \frac{\text{wins}(S_i, S_j)}{\text{wins}(S_i, S_j) + \text{wins}(S_j, S_i)}$$

This metric computes an average relative frequency of wins, excluding ties, and was used in WMT12 and WMT13 (Callison-Burch et al., 2012; Bojar et al., 2013).

The decision to exclude ties isn't without its problems; for example, an evaluation where two systems are nearly always judged equivalent should be relevant in producing the final ranking of systems. Furthermore, as Hopkins and May (2013) point out, throwing out data to avoid biasing a model suggests a problem with the model. We now turn to a description of their model, which addresses these problems.

[1] $\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{o.w.} \end{cases}$

2.2 The Hopkins and May (2013) model

Recent papers (Koehn, 2012; Hopkins and May, 2013) have proposed models focused on the *relative ability* of the competition systems. These approaches assume that each system has a mean quality represented by a Gaussian distribution with a fixed variance shared across all systems. In the graphical model formulation of Hopkins and May (2013), the pairwise judgments (i, S_1, S_2, π) are imagined to have been generated according to the following process:

- Select a source sentence i
- Select two systems S_1 and S_2. A system S_j is associated with a Gaussian distribution $\mathcal{N}(\mu_{S_j}, \sigma_a^2)$, samples from which represent the quality of translations
- Draw two "translations", adding random Gaussian noise with variance σ_{obs}^2 to simulate the subjectivity of the task and the differences among annotators:
$$q_1 \sim \mathcal{N}(\mu_{S_1}, \sigma_a^2) + \mathcal{N}(0, \sigma_{obs}^2)$$
$$q_2 \sim \mathcal{N}(\mu_{S_2}, \sigma_a^2) + \mathcal{N}(0, \sigma_{obs}^2)$$
- Let d be a nonzero real number that defines a fixed decision radius. Produce a rating π according to:[2]
$$\pi = \begin{cases} < & q_1 - q_2 > d \\ > & q_2 - q_1 > d \\ = & \text{otherwise} \end{cases}$$

The task is to then infer the posterior parameters, given the data: the system means μ_{S_j} and, by necessity, the latent values $\{q_i\}$ for each of the pairwise comparison training instances. Hopkins and May do not publish code or describe details of this algorithm beyond mentioning Gibbs sampling, so we used our own implementation,[3] and describe it here for completeness.

After initialization, we have training instances of the form $(i, S_1, S_2, \pi_R, q_1, q_2)$, where all but the q_i are observed. At a high level, the sampler iterates over the training data, inferring values of q_1 and q_2 for each annotation together in a single step of the sampler from the current values of the systems means, $\{\mu_j\}$.[4] At the end of each iteration,

[2] Note that better systems have *higher* relative abilities $\{\mu_{S_j}\}$. Better translations subsequently have on-average higher values $\{q_i\}$, which translate into a *lower* ranking π.

[3] `github.com/keisks/wmt-trueskill`

[4] This worked better than a version of the sampler that changed one at a time.

these means are then recomputed by re-averaging all values of $\{q_i\}$ associated with that system. After the burn-in period, the μs are stored as samples, which are averaged when the sampling concludes.

During each iteration, q_1 and q_2 are resampled from their corresponding system means:

$$q_1 \sim \mathcal{N}(\mu_{S_1}, \sigma_a^2)$$
$$q_2 \sim \mathcal{N}(\mu_{S_2}, \sigma_a^2)$$

We then update these values to respect the annotation π as follows. Let $t = q_1 - q_2$ (S_1 is the winner by human judgments), and ensure that the values are outside the decision radius, d:

$$q_1' = \begin{cases} q_1 & t \geq d \\ q_1 + \frac{1}{2}(d-t) & \text{otherwise} \end{cases}$$

$$q_2' = \begin{cases} q_2 & t \geq d \\ q_2 - \frac{1}{2}(d-t) & \text{otherwise} \end{cases}$$

In the case of a tie:

$$q_1' = \begin{cases} q_1 + \frac{1}{2}(d-t) & t \geq d \\ q_1 & t < d \\ q_1 + \frac{1}{2}(-d-t) & t \leq -d \end{cases}$$

$$q_2' = \begin{cases} q_2 - \frac{1}{2}(d-t) & t \geq d \\ q_2 & t < d \\ q_2 - \frac{1}{2}(-d-t) & t \leq -d \end{cases}$$

These values are stored for the current iteration and averaged at its end to produce new estimates of the system means. The quantity $d - t$ can be interpreted as a *loss function*, returning a high value when the observed outcome is unexpected and a low value otherwise (Figure 1).

2.3 TrueSkill

Prior to 2012, the WMT organizers included reference translations among the system comparisons. These were used as a control against which the evaluators could be measured for consistency, on the assumption that the reference was almost always best. They were also included as data points in computing the system ranking. Another of Bojar et al. (2011)'s suggestions was to exclude this data, because systems compared more often against the references suffered unfairly. This can be further generalized to the observation that not all competitions are equal, and a good model should incorporate some notion of "match difficulty" when evaluating system's abilities. The inference procedure above incorporates this notion implicitly in the inference procedure, but the model itself does not include a notion of match difficulty or outcome surprisal.

A model that does is TrueSkill[5] (Herbrich et al., 2006). TrueSkill is an adaptive, online system that also assumes that each system's skill level follows a Gaussian distribution, maintaining a mean μ_{S_j} for each system S_j representing its current estimate of that system's native ability. However, it also maintains a per-system variance, $\sigma^2_{S_j}$, which represents TrueSkill's uncertainty about its estimate of each mean. After an outcome is observed (a game in which the result is a win, loss, or draw), the size of the updates is proportional to how surprising the outcome was, which is computed from the current system means and variances. If a translation from a system with a high mean is judged better than a system with a greatly lower mean, the result is not surprising, and the update size for the corresponding system means will be small. On the other hand, when an upset occurs in a competition, the means will receive larger updates.

Before defining the update equations, we need to be more concrete about how this notion of surprisal is incorporated. Let $t = \mu_{S_1} - \mu_{S_2}$, the difference in system relative abilities, and let ϵ be a fixed hyper-parameter corresponding to the earlier decision radius. We then define two loss functions of this difference for wins and for ties:

$$v_{\text{win}}(t, \epsilon) = \frac{\mathcal{N}(-\epsilon + t)}{\Phi(-\epsilon + t)}$$
$$v_{\text{tie}}(t, \epsilon) = \frac{\mathcal{N}(-\epsilon - t) - \mathcal{N}(\epsilon - t)}{\Phi(\epsilon - t) - \Phi(-\epsilon - t)}$$

where $\Phi(x)$ is the cumulative distribution function and the $\mathcal{N}$s are Gaussians. Figures 1 and 2 display plots of these two functions compared to the Hopkins and May model. Note how v_{win} (Figure 1) increases exponentially as μ_{S_2} becomes greater than the (purportedly) better system, μ_{S_1}.

As noted above, TrueSkill maintains not only estimates $\{\mu_{S_j}\}$ of system abilities, but also system-specific confidences about those estimates

[5]The goal of this section is to provide an intuitive description of TrueSkill as adapted for WMT manual evaluations, with enough detail to carry the main ideas. For more details, please see Herbrich et al. (2006).

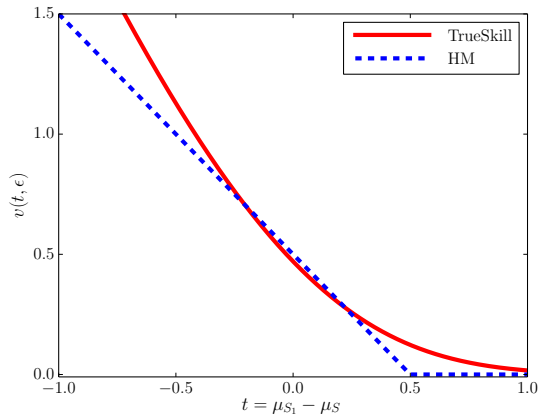

Figure 1: TrueSkill's v_{win} and the corresponding *loss function* in the Hopkins and May model as a function of the difference t of system means ($\epsilon = 0.5, c = 0.8$ for TrueSkill, and $d = 0.5$ for Hopkins and May model).

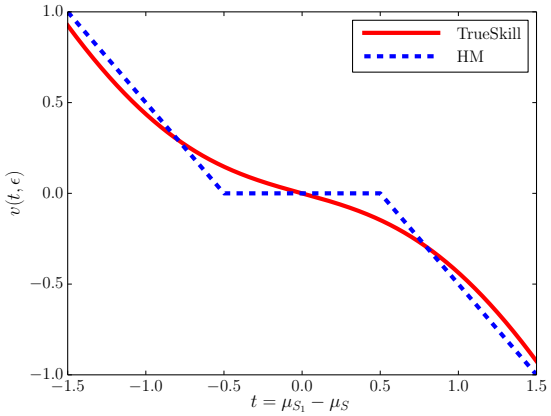

Figure 2: TrueSkills v_{tie} and the corresponding *loss function* in the Hopkins and May model as a function of the difference t of system means ($\epsilon = 0.5, c = 0.3$, and $d = 0.5$).

$\{\sigma_{S_j}\}$. These confidences also factor into the updates: while surprising outcomes result in larger updates to system means, higher confidences (represented by smaller variances) result in *smaller* updates. TrueSkill defines the following value:

$$c^2 = 2\beta^2 + \sigma_{S_1}^2 + \sigma_{S_2}^2$$

which accumulates the variances along β, another free parameter. We can now define the update equations for the system means:

$$\mu_{S_1} = \mu_{S_1} + \frac{\sigma_{S_1}^2}{c} \cdot v\left(\frac{t}{c}, \frac{\epsilon}{c}\right)$$
$$\mu_{S_2} = \mu_{S_2} - \frac{\sigma_{S_2}^2}{c} \cdot v\left(\frac{t}{c}, \frac{\epsilon}{c}\right)$$

The second term in these equations captures the idea about balancing surprisal with confidence, described above.

In order to update the system-level confidences, TrueSkill defines another set of functions, w, for the cases of wins and ties. These functions are multiplicative factors that affect the amount of change in σ^2:

$$w_{\text{win}}(t,\epsilon) = v_{\text{win}} \cdot (v_{\text{win}} + t - \epsilon)$$
$$w_{\text{tie}}(t,\epsilon) = v_{\text{tie}} + \frac{(\epsilon - t) \cdot \mathcal{N}(\epsilon - t) + (\epsilon + t) \cdot \mathcal{N}(\epsilon + t)}{\Phi(\epsilon - t) - \Phi(-\epsilon - t)}$$

The underlying idea is that these functions capture the outcome surprisal via v. This update always decreases the size of the variances σ^2, which means uncertainty of μ decreases as comparisons go on. With these defined, we can conclude by defining the updates for $\sigma_{S_1}^2$ and $\sigma_{S_2}^2$:

$$\sigma_{S_1}^2 = \sigma_{S_1}^2 \cdot \left[1 - \frac{\sigma_{S_1}^2}{c^2} \cdot w\left(\frac{t}{c}, \frac{\epsilon}{c}\right)\right]$$
$$\sigma_{S_2}^2 = \sigma_{S_2}^2 \cdot \left[1 - \frac{\sigma_{S_2}^2}{c^2} \cdot w\left(\frac{t}{c}, \frac{\epsilon}{c}\right)\right]$$

One final complication not presented here but relevant to adapting TrueSkill to the WMT setting: the parameter β and another parameter (not discussed) τ are incorporated into the update equations to give more weight to recent matches. This "latest-oriented" property is useful in the gaming setting for which TrueSkill was built, where players improve over time, but is not applicable in the WMT competition setting. To cancel this property in TrueSkill, we set $\tau = 0$ and $\beta = 0.025 \cdot |A| \cdot \sigma^2$ in order to lessen the impact of the order in which annotations are presented to the system.

2.4 Data selection with TrueSkill

A drawback of the standard WMT data collection method is that it samples uniformly from the space of pairwise system combinations. This is undesirable: systems with vastly divergent relative ability need not be compared as often as systems that are more evenly matched. Unfortunately, one cannot sample non-uniformly without knowing ahead of time which systems are better. TrueSkill provides a solution to this dilemma with its *match-selection* ability: systems with similar means and low variances can be confidently considered to be close matches. This presents a strong possibility of reducing the amount of data that needs to be

collected in the WMT competitions. In fact, the TrueSkill formulation provides a way to compute the probability of a draw between two systems, which can be used to compute for a system S_i a conditional distribution over matches with other systems $\{S_{j\neq i}\}$.

Formally, in the TrueSkill model, the *match-selection* (chance to draw) between two players (systems in WMT) is computed as follows:

$$p_{\text{draw}} = \sqrt{\frac{2\beta^2}{c^2}} \cdot \exp(-\frac{(\mu_a - \mu_b)^2}{2c^2})$$

However, our setting for canceling the "latest-oriented" property affects this matching quality equation, where most systems are almost equally competitive (≈ 1). Therefore, we modify the equation in the following manner which simply depends on the difference of μ.

$$\hat{p}_{\text{draw}} = \frac{1}{\exp(|\mu_a - \mu_b|)}$$

TrueSkill selects the matches it would like to create, according to this selection criteria. We do this according to the following process:

1. Select a system S_1 (e.g., the one with the highest variance)
2. Compute a normalized distribution over matches with other systems pairs $\hat{p}_{\text{draw}}$
3. Draw a system S_2 from this distribution
4. Draw a source sentence, and present to the judge for annotation

3 Experimental setup

3.1 Datasets

We used the evaluation data released by WMT13.[6] The data contains (1) five-way system rankings made by either researchers or Turkers and (2) translation data consisting of source sentences, human reference translations, and submitted translations. Data exists for 10 language pairs. More details about the dataset can be found in the WMT 2013 overview paper (Bojar et al., 2013).

Each five-way system ranking was converted into ten pairwise judgments (§2). We trained the models using randomly selected sets of 400, 800, 1,600, 3,200, and 6,400 pairwise comparisons, each produced in two ways: selecting from all researchers, or split between researchers and Turkers. An important note is that the training data differs according to the model. For the Expected Wins and Hopkins and May model, we simply sample uniformly at random. The TrueSkill model, however, selects its own training data (with replacement) according to the description in Section 2.4.[7]

For tuning hyperparameters and reporting test results, we used development and test sets of 2,000 comparisons drawn entirely from the researcher judgments, and fixed across all experiments.

3.2 Perplexity

We first compare the Hopkins and May model and TrueSkill using perplexity on the test data T, computed as follows:

$$\text{ppl}(p|T) = 2^{-\sum_{(i,S_1,S_2,\pi)\in T} \log_2 p(\pi|S_1,S_2)}$$

where p is the model under consideration. The probability of each observed outcome π between two systems S_1 and S_2 is computed by taking a difference of the Gaussian distributions associated with those systems:

$$\begin{aligned}\mathcal{N}(\mu_\delta, \sigma_\delta^2) &= \mathcal{N}(\mu_{S_1}, \sigma_{S_1}^2) - \mathcal{N}(\mu_{S_2}, \sigma_{S_2}^2) \\ &= \mathcal{N}(\mu_{S_1} - \mu_{S_2}, \sigma_{S_1}^2 + \sigma_{S_2}^2)\end{aligned}$$

This Gaussian can then be carved into three pieces: the area where S_1 loses, the middle area representing ties (defined by a decision radius, r, whose value is fit using development data), and a third area representing where S_1 wins. By integrating over each of these regions, we have a probability distribution over these outcomes:

$$p(\pi \mid S_1, S_2) = \begin{cases} \int_{-\infty}^{0} \mathcal{N}(\mu_\delta, \sigma_\delta^2) & \text{if } \pi \text{ is } > \\ \int_{0}^{r} \mathcal{N}(\mu_\delta, \sigma_\delta^2) & \text{if } \pi \text{ is } = \\ \int_{r}^{\infty} \mathcal{N}(\mu_\delta, \sigma_\delta^2) & \text{if } \pi \text{ is } < \end{cases}$$

We do not compute perplexity for the Expected Wins model, which does not put any probability mass on ties.

[6] statmt.org/wmt13/results.html

[7] We use a Python implementation of TrueSkill (github.com/sublee/trueskill).

3.3 Accuracy

Perplexity is often viewed as a neutral metric, but without access to unbounded training data or the true model parameters, it can only be approximated. Furthermore, it does not always correlate perfectly with evaluation metrics. As such, we also present accuracy results, measuring each model's ability to predict the values of the ternary pairwise judgments made by the annotators. These are computed using the above equation, picking the highest value of $p(\pi)$ for all annotations between each system pair (S_i, S_j). As with perplexity, we emphasize that these predictions are functions of the system pair only, and not the individual sentences under consideration, so the same outcome is always predicted for all sentences between a system pair.

3.4 Parameter Tuning

We follow the settings described in Hopkins and May (2013) for their model: $\sigma_a = 0.5, \sigma_{\text{obs}} = 1.0$, and $d = 0.5$. In TrueSkill, in accordance with the Hopkins and May model, we set the initial μ and σ values for each system to 0 and 0.5 respectively, and ϵ to 0.25.

For test data, we tuned the "decision radius" parameter r by doing grid search over $\{0.001, 0.01, 0.1, 0.3, 0.5\}$, searching for the value which minimized perplexity and maximized accuracy on the development set. We do this for each model and language pair. When tuned by perplexity, r is typically either 0.3 or 0.5 for both models and language pairs, whereas, for accuracy, the best r is either 0.001, 0.01, or 0.1.

4 Results

4.1 Model Comparison

Figure 3 shows the perplexity of the two models with regard to the number of training comparisons. The perplexities in the figure are averaged over all ten language pairs in the WMT13 dataset. Overall, perplexities decrease according to the increase of training size. The Hopkins and May and TrueSkill models trained on both researcher and Turker judgments are comparable, whereas the Hopkins and May model trained on researcher judgments alone shows lower perplexity than the corresponding TrueSkill model.

In terms of accuracy, we see that the TrueSkill model has the highest accuracies, saturating at just over 3,000 training instances (Figure 4). TrueSkill

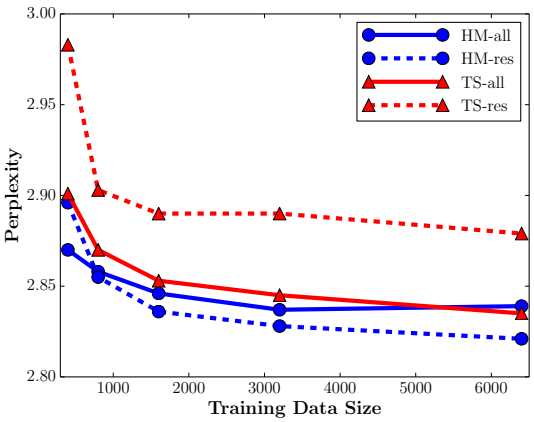

Figure 3: Model Perplexities for WMT13 dataset. 'all' indicates that models are trained on both researcher and Turker judgements, and 'res' means that models are trained on only researcher judgements.

outperforms Expected Win and the Hopkins and May, especially when the training size is small (Table 2). We also note that training on researcher judgments alone (dashed lines) results in better performance than training on both researchers and Turker judgments. This likely reflects both a better match between training and test data (recall the test data consists of researcher judgments only), as well as the higher consistency of this data, as evidenced by the annotator agreement scores published in the WMT overview paper (Bojar et al., 2013). Recall that the models only have access to the system pair (and not the sentences themselves), and thus make the same prediction for π for a particular system pair, regardless of which source sentence was selected. As an upper bound for performance on this metric, Table 2 contains an oracle score, which is computed by selecting, for each pair of systems, the highest-probability ranking.[8]

Comparing the plots, we see there is not a perfect relationship between perplexity and accuracy among the models; the low perplexity does not mean the high accuracy, and in fact the order of the systems is different.

4.2 Free-for-all matches

TrueSkill need not deal with judgments in pairs only, but was in fact designed to be used in a variety of settings, including N-way free-for-all games

[8] Note that this might not represent a consistent ranking among systems, but is itself an upper bound on the highest-scoring consistent ranking.

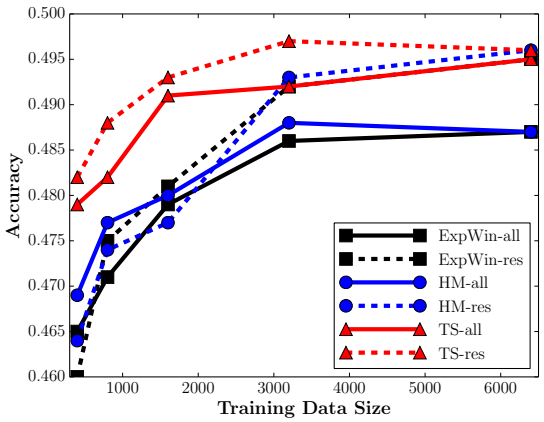

Figure 4: Model accuracies with different training domain for WMT13 dataset.

Train Size		Exp-Win	HM	TrueSkill
all	400	0.465	0.471	**0.479**
	800	0.471	0.475	**0.483**
	1600	0.479	0.477	**0.493**
	3200	0.486	0.489	**0.493**
	6400	0.487	0.490	**0.495**
res	400	0.460	0.463	**0.484**
	800	0.475	0.473	**0.488**
	1600	0.481	0.482	**0.493**
	3200	0.492	0.494	**0.497**
	6400	0.495	0.496	**0.497**
Upper Bound			0.525	

Table 2: Model accuracies: models are tuned by accuracy instead of perplexity. Upper bound is computed by selecting the most frequent choice ($<, >, =$) for each system pair.

with many players all competing for first place. This adapts nicely to WMT's actual collection setting. Recall that annotators are presented with five translations which are then ranked; we can treat this setting as a 5-way free-for-all match. While the details of these updates are beyond the scope of this paper, they are presented in the original model and are implemented in the toolkit we used. We thus also conducted experiments varying the value of N from 2 to 5.

The results are shown in Tables 3 and 4, which hold constant the number of matches and pairwise judgments, respectively. When fixing the number of matches, the 5-way setting is at an advantage, since there is much more information in each match; in contrast, when fixing the number of pairwise comparisons, the 5-way setting is at a disadvantage, since many fewer competitions constitute these comparisons. The results bear this out, but also suggest that the standard WMT setting — which extracts ten pairwise comparisons from each 5-way match and treats them independently — works well. We will not speculate further here, but provide this experiment purely to motivate potential future work. Here we will focus our conclusions to the pair-wise ranking scenario.

#	N=2	N=3	N=4	N=5
400	0.479	0.482	0.491	**0.492**
800	0.483	0.493	0.495	**0.495**
1600	0.493	0.492	**0.497**	0.495
3200	0.493	0.494	**0.498**	0.497
6400	0.495	**0.498**	0.498	0.498

Table 3: Accuracies when training with N-way free-for-all models, fixing the number of matches.

#	N=2	N=3	N=4	N=5
400	**0.479**	0.475	0.470	0.459
800	0.483	**0.488**	0.476	0.466
1600	**0.493**	0.488	0.481	0.481
3200	**0.493**	0.492	0.487	0.489
6400	0.495	**0.496**	0.494	0.495

Table 4: Accuracies when training with N-way free-for-all models, fixing the number of pairwise comparisons.

5 Reduced Data Collection with Non-uniform Match Selection

As mentioned earlier, a drawback of the selection of training data for annotation is that it is sampled uniformly from the space of system pair competitions, and an advantage of TrueSkill is its ability to instead compute a distribution over pairings and thereby focus annotation efforts on competitive matches. In this section, we report results in the form of heat maps indicating the percentage of pairwise judgments requested by TrueSkill across the full cross-product of system pairs, using the WMT13 French-English translation task.

Figure 5 depicts a system-versus-system heat map for all judgments in the dataset. Across this figure and the next two, systems are sorted along each axis by the final values of μ inferred by TrueSkill during training, and the heat of each square is proportional to the percentage of judgments obtained between those two systems. The diagonal reflects the fact that systems do not compete against themselves, and the stripe at row and column 5 reflects a system that was entered late

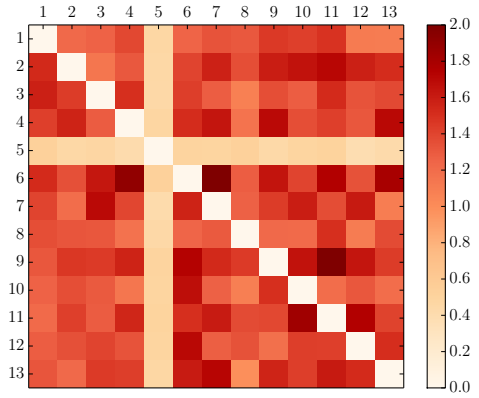

Figure 5: Heat map for the ratio of pairwise judgments across the full cross-product of systems in the WMT13 French-English translation task.

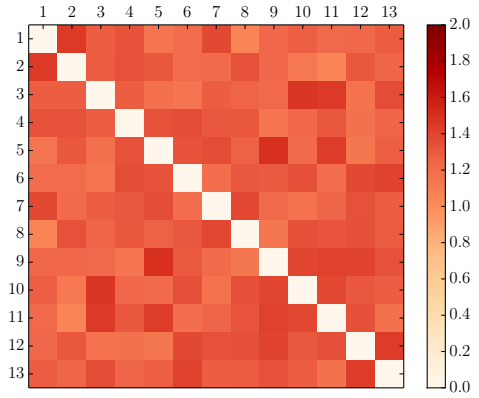

Figure 6: Heat map for the ratio of pairwise judgments across the full cross-product of systems used in the *first* 20% of TrueSkill model.

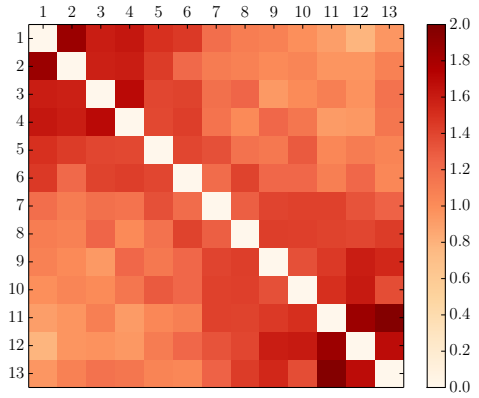

Figure 7: Heat map for the ratio of pairwise judgments across the full cross-product of systems used in the *last* 20% of TrueSkill model.

into the WMT13 competition and thus had many fewer judgments. It is clear that these values are roughly uniformly distributed. This figure serves as a sort of baseline, demonstrating the lack of patterns in the data-selection process.

The next two figures focus on the data that TrueSkill itself selected for its use from among all of the available data. Figure 6 is a second heat map presenting the set of system pairs selected by TrueSkill for the *first* 20% of its matches chosen during training, while Figure 7 presents a heat map of the *last* 20%. The contrast is striking: whereas the judgments are roughly uniformly distributed at the beginning, the bulk of the judgments obtained for the last set are clustered along the diagonal, where the most competitive matches lie.

Together with the higher accuracy of TrueSkill, this suggests that it could be used to decrease the amount of data that needs to be collected in future WMT human evaluations by focusing the annotation effort on more closely-matched systems.

6 Clustering

As pointed out by Koehn (2012), a ranking presented as a total ordering among systems conceals the closeness of comparable systems. In the WMT13 competition, systems are grouped into clusters, which is equivalent to presenting only a *partial* ordering among the systems. Clusters are constructed using bootstrap resampling to infer many system rankings. From these rankings, *rank ranges* are then collected, which can be used to construct 95% confidence intervals, and, in turn, to cluster systems whose ranges overlap. We use a similar approach for clustering in the TrueSkill model. We obtain rank ranges for each system by running the TrueSkill model 100 times,[9] throwing out the top and bottom 2 rankings for each system, and clustering where rank ranges overlap. For comparison, we also do this for the other two models, altering the amount of training data from 1k to 25k in increments of 1,000, and plotting the number of clusters that can be obtained from each technique on each amount of training data.

Figure 8 show the number of clusters according to the increase of training data for three models. TrueSkill efficiently split the systems into clusters compared to other two methods. Figure 9 and 10 present the result of clustering two different size of

[9] We also tried the sampling 1,000 times and the clustering granularities were the same.

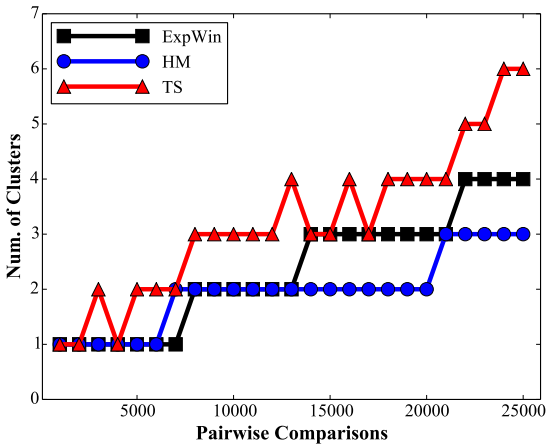

Figure 8: The number of clusters according to the increase of training data for WMT13 French-English (13 systems in total).

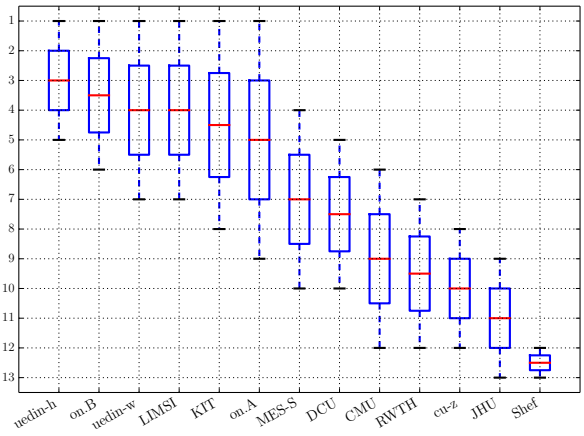

Figure 9: The result of clustering by TrueSkill model with 1K training data from WMT13 French-English. The boxes range from the lower to upper quartile values, with means in the middle. The whiskers show the full range of each system's rank after the bootstrap resampling.

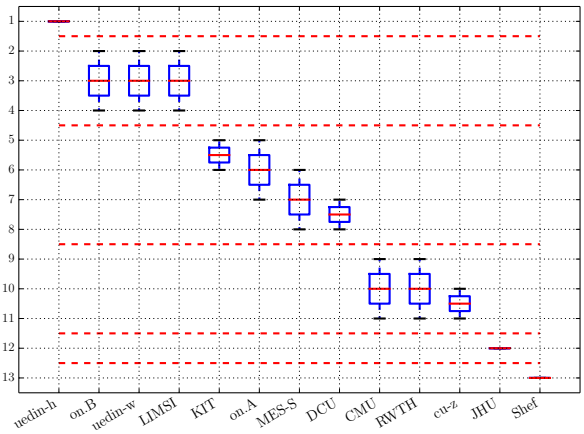

Figure 10: The result of clustering by TrueSkill model with 25K training data. Dashed lines separate systems with non-overlapping rank ranges, splitting the data into clusters.

training data (1K and 25K pairwise comparisons) on the TrueSkill model, which indicates that the rank ranges become narrow and generate clusters reasonably as the number of training samples increases. The ranking and clusters are slightly different from the official result (Table 1) mainly because the official result is based on Expected Wins.

One noteworthy observation is that the ranking of systems between Figure 9 and Figure 10 is the same, further corroborating the stability and accuracy of the TrueSkill model even with a small amount of data. Furthermore, while the need to cluster systems forces the collection of significantly more data than if we wanted only to report a total ordering, TrueSkill here produces nicely-sized clusters with only 25K pairwise comparisons, which is nearly one-third large of that used in the WMT13 campaign (80K for French-English, yielding 8 clusters).

7 Conclusion

Models of "relative ability" (Koehn, 2012; Hopkins and May, 2013) are a welcome addition to methods for inferring system rankings from human judgments. The TrueSkill variant presented in this paper is a promising further development, both in its ability to achieve higher accuracy levels than alternatives, and in its ability to sample non-uniformly from the space of system pair matchings. It's possible that future WMT evaluations could significantly reduce the amount of data they need to collect, also potentially allowing them to draw from expert annotators alone (the developers of the participating systems), without the need to hire non-experts on Mechanical Turk.

One piece missing from the methods explored and proposed in this paper is models of the actual translations being compared by judges. Clearly, it is properties of the sentences themselves that judges use to make their judgments, a fact which is captured only indirectly by modeling translation qualities sampled from system abilities. This observation has been used in the development of automatic evaluation metrics (Song and Cohn, 2011), and is something we hope to explore in future work for system ranking.

References

Ondřej Bojar, Miloš Ercegovčević, Martin Popel, and Omar Zaidan. 2011. A Grain of Salt for the WMT Manual Evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. TrueSkill™: A Bayesian Skill Rating System. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, pages 569–576, Vancouver, British Columbia, Canada, December. MIT Press.

Mark Hopkins and Jonathan May. 2013. Models of translation competitions. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1416–1424, Sofia, Bulgaria, August. Association for Computational Linguistics.

Philipp Koehn. 2012. Simulating Human Judgment in Machine Translation Evaluation Campaigns. In *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*, pages 179–184, Hong Kong, China, December. International Speech Communication Association.

Adam Lopez. 2012. Putting Human Assessments of Machine Translation Systems in Order. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 1–9, Montréal, Canada, June. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, July. Association for Computational Linguistics.

Xingyi Song and Trevor Cohn. 2011. Regression and Ranking based Optimisation for Sentence Level MT Evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 123–129, Edinburgh, Scotland, July. Association for Computational Linguistics.