Lightly Supervised Quality Estimation

Matthias Sperber[1,⋆], **Graham Neubig**[2], **Jan Niehues**[1], **Sebastian Stüker**[1], **Alex Waibel**[1]
[1]Karlsruhe Institute of Technology, Germany
[2]Carnegie Mellon University, USA
⋆matthias.sperber@kit.edu

Abstract

Evaluating the quality of output from language processing systems such as machine translation or speech recognition is an essential step in ensuring that they are sufficient for practical use. However, depending on the practical requirements, evaluation approaches can differ strongly. Often, reference-based evaluation measures (such as BLEU or WER) are appealing because they are cheap and allow rapid quantitative comparison. On the other hand, practitioners often focus on manual evaluation because they must deal with frequently changing domains and quality standards requested by customers, for which reference-based evaluation is insufficient or not possible due to missing in-domain reference data (Harris et al., 2016). In this paper, we attempt to bridge this gap by proposing a framework for lightly supervised quality estimation. We collect manually annotated scores for a small number of segments in a test corpus or document, and combine them with automatically predicted quality scores for the remaining segments to predict an overall quality estimate. An evaluation shows that our framework estimates quality more reliably than using fully automatic quality estimation approaches, while keeping annotation effort low by not requiring full references to be available for the particular domain.

1 Introduction

Quality evaluation is a key requirement for developing and employing language technology, enabling users and engineers to judge overall quality of the output, detect key problems, improve systems, and choose among competing systems. Although most users and engineers share these goals, the chosen evaluation approaches can differ strongly, with some people resorting to automatic, reference-based evaluation, while others rely on manual evaluation for their purposes. This is especially pronounced in the case of machine translation (MT), as pointed out by Harris et al. (2016). On one hand, much research effort has been devoted to devising reference-based methods such as BLEU (Papineni et al., 2002) that are well-correlated with human judgment. On the other hand, practitioners need to react to changing domains from customer to customer, and reflect multi-faceted quality requirements that are difficult to measure in a single, generic score, often leaving manual evaluation as the only choice.

In recent years, automatic quality estimation (QE) has emerged as a method that could potentially address the lack of flexibility of reference-based evaluation to deal with changing requirements, and the high effort of manual evaluation. Automatic QE uses machine learning techniques that are trained on a dataset of output-quality pairs in order to predict the quality of some new system output. It can be used to predict arbitrary quality metrics, provided that suitably labeled training data is available. However, in practice automatic QE has been found difficult and not reliable enough when applied to new domains or unknown systems, e.g. in MT (de Souza et al., 2015a) and automatic speech recognition (ASR) (Negri et al., 2014).

In this work, we explore a middle ground between automatic QE and manual evaluation, aiming to allow the QE system to adapt to the particular test data under consideration while keeping manual effort at an affordable level. We refer to this approach as *lightly supervised*[1] QE. Our general approach is

[1] In this paper, the terms supervised and unsupervised refer to whether or not manual annotation is necessary at *test* time.

to divide the test data into smaller segments such as sentences or utterances, obtain automatic quality estimates for all these segments, and manually annotate only a small number of segments. We then compute aggregated quality estimates by averaging over segment-level scores, and explore two ways to aggregate manual and automatic scores.

The first aggregation approach computes the overall quality by averaging over *only the automatic* segment scores, using the manual annotations to adapt the automatic QE regressor. The method has the potential to improve predictions by reflecting not only the topical domain, but also the particular ASR or MT system, and even use-case specific quality standards.

The second aggregation approach computes quality by averaging over only the collected *manual segment scores*, using the automatic scores to choose what segments should be manually evaluated. In particular, we expect segments that are predicted to be close to the average across segments to be most indicative of overall quality. Instead of QE predictions, this aggregation approach can also directly exploit confidence scores from the decoder, eliminating the need for training a QE regressor.

We evaluate our approach for a variety of situations, focusing on the output of MT and ASR systems. We find that for MT, we can achieve a desired accuracy of quality estimation with much less effort compared to fully manual annotation. For instance, when annotating 100 words, in an in-domain setting we reduce the error by 22% and 19% relative over a fully automatic and a fully manual baseline. In an out-of-domain setting, relative improvements are 10% and 71%. For ASR, we obtain no improvements using the regression approach, but obtain promising results when using confidence scores instead: At 100 annotated words, the relative improvements are 59% and 54%.

2 Lightly Supervised Estimation Framework

This section outlines our lightly supervised estimation framework. The input to the framework is the hypothesized translation (transcription) of some MT (ASR) system given some particular input document (audio). We will refer to these as *hypothesis*, *system*, and *document* throughout the paper. Here, the document can be any form of test corpus that is representative of our targeted application, such as a document written in a particular style, data from one or several speakers, or a topical domain. Our goal is to estimate the average quality of the generated target sentences for a particular document with respect to some evaluation measure. The evaluation measure can be chosen arbitrarily, the only requirement being that it can be assigned on a per-sentence level. Possible examples are translation edit rate (TER) or sentence-level BLEU for MT, word error rate (WER) for ASR, or human ratings. We assume that the hypothesis is segmented in some way. The choice of segmentation is arbitrary, but sentence or utterance boundaries are a natural choice.

For purposes of this paper, we define document-level quality as the weighted average of the segment-level scores. Let ALL denote the index set of all segments in our document, y_i and w_i the true quality and weight of the i-th segment. The overall quality $Q^{(\text{true})}$ to be estimated is defined as:

$$Q^{(\text{true})} := \sum_{i \in \text{ALL}} w_i y_i \tag{1}$$

Note that this definition does not aim to handle document-level discourse phenomena such as coherence, cohesion, and and consistency, but estimates the average sentence-level quality for the document in order to evaluate the overall quality of the whole output hypothesis. In this paper, we weigh segments proportionally to their length, although future work may investigate more sophisticated notions of segment importance. Our definition of document quality is simplistic, but widely used in both MT and ASR communities, e.g. document- or corpus-level WER, TER, METEOR (Banerjee and Lavie, 2005), and human rankings are usually computed this way. BLEU is computed on the corpus level and thus not directly usable with our approach, but we can instead resort to computing average sentence-level BLEU variants such as BLEU+1 (Lin and Och, 2004) that essentially differ only in the smoothing details.

Our lightly supervised estimation framework determines document quality in several steps. The first step is to automatically estimate the quality for each segment in the hypothesis (§4). This can be achieved by training a regressor with the desired target measure, as discussed in numerous previous works. The second step is then to manually annotate the quality score for a certain number of segments. In our evaluation (§5), we experiment with typical amounts of tens to hundreds of annotated words. The final step is to aggregate manual and automatic scores into a document-level estimate (§3).

3 Aggregation of Manual and Automatic Scores

We assume for a moment that we know how to collect automatic and manual segment-level scores, and discuss how they may be aggregated into a document-level estimate. Let $p_i, \forall i \in \textsc{all}$ denote the automatically predicted scores for all segments in the document, corresponding to the index set $\textsc{all}$. Let $y_i, \forall i \in \textsc{sel}$ denote the manually annotated segment scores for a subset of all segments in a document, with indices $\textsc{sel} \subseteq \textsc{all}$. w_i is the number of tokens in the system output for the i-th segment, and $Z^{(\mathcal{S})}$ is the total length of all segments indexed by $\mathcal{S}$. $Q^{(\text{true})}$ is the true document quality to be estimated.

3.1 Baselines

First, consider a *fully manual* baseline: Here, we randomly select segments from $\textsc{all}$ to create $\textsc{sel}$, and use the weighted average of manually annotated quality scores over these segments. Note that even though the segment-level scores y_i are reliable, the aggregate $Q^{(\text{man})}$ will be inaccurate if we keep the size of the annotated sample $\textsc{sel}$ small, as desired:

$$Q^{(\text{man})} = \frac{1}{Z^{(\textsc{sel})}} \sum_{i \in \textsc{sel}} w_i y_i \tag{2}$$

In contrast, the *fully automatic* baseline uses only the automatically predicted scores. Because automatic scores are available for all segments, the sample size is much bigger. On the other hand, the regressor used for prediction is often biased to predicting values p_i close to those indicated by the training data, which may differ considerably from the true values for y_i, especially for documents that are less similar to the training data. The formula is as follows:

$$Q^{(\text{auto})} = \frac{1}{Z^{(\textsc{all})}} \sum_{i \in \textsc{all}} w_i p_i \tag{3}$$

3.2 Bias vs. Variance: Why Aggregation is Challenging

Our goal is to improve over these baselines, and to motivate our methods we first frame these baselines in the context of the bias-variance tradeoff widely discussed in machine learning (Hastie et al., 2009). Here, the $\text{bias} = \mathbf{E}\big[Q^{(\text{est})} - Q^{(\text{true})}\big]^2$ describes the (squared) average error of the document-level estimate, with the expectation averaging over the randomness in the selection of annotated segments $\textsc{sel}$, and other factors that one would consider random, such as particular training data for automatic QE, or randomness in the document that one might want to abstract from. The $\text{variance} = \mathbf{E}\big[\big(Q^{(\text{est})} - \mathbf{E}[Q^{(\text{est})}]\big)^2\big]$ describes by how much the estimates vary from one another, by again taking into account the randomness in the selection of segments, data, etc.

Based on these definitions, we observe that $Q^{(\text{man})}$ is subject to high variance, because the small sample size makes it sensitive to the randomness of the data. On the other hand, its bias is zero, since averaging over all randomness would cancel out estimation errors exactly. $Q^{(\text{auto})}$, on the other hand, has low variance because it always considers the whole document, but high bias for documents that are less similar to the data used to train the regressor. Figure 1 illustrates this interplay.

A straight-forward way to combine automatic and manual scores would be some form of interpolation:

$$Q^{(\text{interp})} = \alpha Q^{(\text{man})} + (1 - \alpha) Q^{(\text{auto})} \tag{4}$$

We experimented with several interpolation approaches in this spirit, but found that they do not work, because the high bias of the automatic estimate hurts accuracy more than could be compensated for by reduced variance of the interpolated estimate. The following subsections describe our refined aggregation strategies, which are more suited to solving this problem.

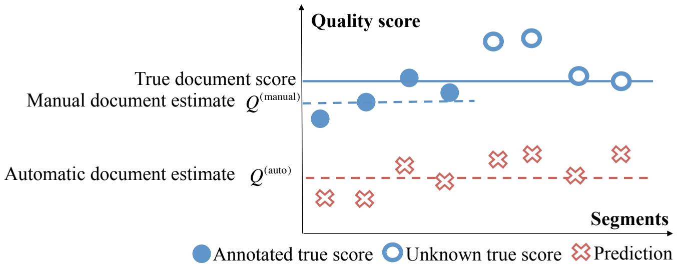


Figure 1: Schematic interplay between annotated and predicted estimates. The automatic aggregate has low variance because it includes all segments, but regressor-training data mismatch can lead to severely biased estimates. On the other hand, the manual annotations are unbiased per definition, but the potentially small number of samples causes high variance in the aggregate.

3.3 Proposed Strategy 1: Regressor Adaptation

In our first strategy, we use the annotated segments to adapt the regressor. This is done by casting the aggregation as a domain adaptation problem, in which we treat the original training data as background data, and the annotated segments of our document as in-domain data. Details for how the domain adaptation is performed are given in §4.1. As in the fully manual baseline, segments for annotation are selected at random. The document estimate is as in the fully automatic baseline, except that we now use the adapted regressor to produce the automatic scores $p_i^{(\text{adapt})}$. The hope is that the adaptation will reduce the regressor bias by shifting predictions closer to the document mean. The formula is as below:

$$Q^{(\text{adapt})} = \frac{1}{Z^{(\text{ALL})}} \sum_{i \in \text{ALL}} w_i p_i^{(\text{adapt})} \tag{5}$$

3.4 Proposed Strategy 2: Active Selection of Segments for Annotation

This approach attempts to select segments for manual annotation that are representative of the whole document. As a proxy for representativeness, we compute how close a segment's predicted quality is to the median prediction across the document. This choice is motivated by our definition of overall quality as the averaged segment-level quality. Specifically, we sort the segments by their automatic scores, and add the median[2] segment to SEL, the set of segments to annotate. Then, we add an equal number of segments to the left and to right according to this ordering, until the desired number of segments is reached. The document estimate is then calculated in the same manner as Equation 2.

The hope is that the active selection of annotated segments will reduce the sample variance. Note that unlike the first strategy, this one does not restrict the metric of the automatic scores to correspond to the target evaluation metric, because p_i does not appear directly in the sum. For instance, it would be possible to use segment-level BLEU scores or even confidence scores to compute SEL, despite our evaluation target being TER or human rating. The only requirement is that the predicted scores correlate with the target metric to some extent.

4 Automatic Segment Scores

4.1 Regression-based Approach

The first class of segment-level automatic scores we consider are QE scores obtained by a black box regressor that uses only surface features derived from the system input and output. It has the advantage of being agnostic to the interiors of the system that was used, and being able to predict any desired metric.

Features: We follow previous works for feature extraction. For MT, we use the 17 baseline features from the WMT QE shared task (Bojar et al., 2015), extracted using the QuEst toolkit (Specia et al., 2013). For ASR, we extracted signal-, hybrid-, and textual features as described in (Negri et al., 2014).

[2] We found the median criterion to outperform a mean criterion and other alternatives.

Regression Algorithm: We use extremely randomized trees (XTs) (Geurts et al., 2006), following Negri et al. (2014). XTs are a tree ensemble with a high amount of randomization, which has been reported to effectively reduce prediction variance, and can be trained quickly. Negri et al. (2014) report slight improvements for XTs over support vector regression (suggested by Bojar et al. (2015) as a baseline). In addition, our preliminary experiments indicate that XTs have more stable adaptation behavior.

Domain Adaptation: Our regressor adaptation aggregation strategy requires adapting the regressor using the collected manual labels. Our adaptation approach is to add binary indicator features for every document and system, including the document and system currently evaluated. Formally, we perform a projection $\Phi^{i,j} : \mathbb{R}^F \rightarrow \mathbb{R}^{F+D+S}$ from the original F-dimensional feature space into an $(F + D + S)$-dimensional augmented feature space. Here, F is the number of original features, D is the number of training documents plus one for the document under test, and S is the number of training systems plus one for the system under test. Further, i is the index of the particular training- or test-document, j is the index of the particular training- or test-system. The projection is defined as

$$\Phi^{ij}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{e}_i, \mathbf{e}_j \rangle \tag{6}$$

with $\mathbf{e}_i$ being the D-dimensional zero-vector with only the i-th position set to one, and $\mathbf{e}_j$ an analogously defined S-dimensional vector.

We hope that this approach enables decoupled learning of the overall quality of a document or system (via the indicator features), and learning to distinguish quality between segments (via the original features). In general, the indicator features permit the regressor more flexibility to deal with the in-domain training samples, whose number is much smaller than that of the background data. We found the indicators to be helpful also when no adaptation is performed, possibly because it allows a form of multitask learning over the different documents and systems. Therefore, we use indicator features for both aggregation strategies. Our indicator features can be seen as a simplification of Daumé III (2007)'s method: here, in a typical adaptation scenario with in-domain and out-of-domain data, all features are *replicated* for both domains. In our case we deal with a potentially large number of domains (for each document and system), and feature replication would greatly enlarge the feature space and risk overfitting. Hence, we deem the proposed indicator features more appropriate for our purposes.

Predicting Deviation from the Document-Mean: Our active selection strategy only makes use of relative differences between segment scores. Therefore, at training time, we replace the label y_{ji} for the i-th segment of the j-th document (system output) by $y_{ji} - Q_j^{(true)}$, where $Q_j^{(true)}$ is the true overall quality of the j-th document. In this way, the regressor only needs to predict by how far (and in what direction) the segment deviates from the document mean, and the training objective becomes more directly meaningful. The indicator features are particular helpful here as they allow the model to learn and factor out the overall document quality. Note that predicting deviations is not useful for the regressor adaptation strategy, because the automatic scores are no longer meaningful in absolute terms.

4.2 Confidence-based Approach

The second class of automatic segment-level scores are confidence scores obtained directly from the system. Confidence scores are better-informed and potentially better correlated with true segment quality. However, unlike the regressor outputs, confidence scores do not share the same unit as the target evaluation metric. It would be possible to add these as a feature to a regressor as above, but at least for the QE training data used in our experiments, confidence scores were not available for all systems.

Instead, we propose using confidence scores as-is, which frees us from having to train a regressor and collecting training data for it. This approach can only be used with the active selection aggregation strategy. In this paper, we use confidence scores only in the ASR setting, and compute segment confidences as average word-level posterior probabilities derived from lattice- or consensus decoding.

5 Experiments

We conduct experiments to answer the following research questions:

Name	Description	# systems	# documents	# segments
MT.in-domain	WMT 2015 submissions, English to German news task	20	81	43380
MT.out-of-domain	WMT 2014 submissions, English to German medical task (khreshmoi testset)	6	1	6000
ASR.in-domain	IWSLT 2013 submissions, English TED task	13	28	28496
ASR.out-of-domain	English KIT lectures, decoded with a standard in-house ASR system	1	6	1849
ASR.conf-task	Mixed ASR data for which confidence scores were available	4	21	2431

Table 1: Data used in our experiments.

- How much effort is needed to produce quality estimates of a certain accuracy?
- Can we reduce effort, compared to the baselines, using our two proposed strategies?
- How effective are the proposed features, such as indicator features, predicting deviations, and using black box regression or confidences?
- Are there difference between in-domain and out-of-domain settings, or between MT and ASR?

We restrict ourselves to reference-based metrics for simplicity, namely TER for MT, and WER for ASR. Our experiments are simulated in the sense that we have reference translations/transcriptions available, and simulate a human annotator who replicates these exactly. Our main evaluation measure is mean absolute error (MAE)[3] between the predicted and true document-level TER/WER.

We use 5 datasets as indicated in Table 1, with testing data varying over all datasets, but only the ones labeled "in-domain" used for regressor training. Moreover, for each evaluated document the QE regressor was retrained with training data excluding that which corresponded to the same system *or* document currently tested (for in-domain tests). This makes even our in-domain scenario more challenging than some of the previous works on automatic QE (Specia et al., 2015).

We use scikit-learn (Pedregosa et al., 2011) for regressor training. We assign weights to training samples proportional to their segment length, because longer segments are weighted more strongly in our aggregation strategies (§3) and are thus more important to be accurately predicted. We perform random search with 20 iterations to optimize hyper-parameters (namely, the max-depth and min-samples-split parameters of XTs) in terms of mean squared error.[4] Tuning is conducted separately for every test document, using 10-fold cross validation on the respective training data. For regressor adaptation, we experimented with different weights for the adaptation samples, but observed only minor gains and decided to weight all data equally for simplicity.

5.1 Evaluation of Automatic Scores

We first analyze the performance of the fully automatic scores in terms of MAE and Pearson linear correlation coefficients. We investigate both segment-level and document-level performance, the latter being identical to the fully automatic baseline (§3.1). For comparison, we evaluate a mean-predictor baseline that always predicts the training mean, regardless of the input features. This baseline has been found surprisingly strong previously (Negri et al., 2014; Specia et al., 2015), which we confirm in Table 2. On segment-level, gains over the mean-predictor baseline are clearly visible only for the ASR setting. As expected, the out-of-domain tasks appear much more difficult than the in-domain setting. Note that even though the mean baseline sometimes achieves lower MAE, the XT regressor maintains the advantages

[3] Graham (2015) argues that correlation is better for evaluating *sentence-level* QE, because MAE can be improved by transformation to match estimated global mean and variance. However, we find MAE more indicative for our purpose as it measures not only how well systems are compared against one another, but also how well overall quality is judged in absolute terms. Moreover, collecting global statistics for transformation seems problematic when flexibility for domain changes is required.

[4] Tuning directly for MAE yielded similar results.

	Segment level			Document level		
	↓MAE	↓MAE	↑Pearson	↓MAE	↓MAE	↑Pearson
	mean	XT	XT	mean	XT	XT
MT.in-domain	**21.0**	21.2	0.13	7.3	**5.8**	0.16
MT.out-of-domain	**14.9**	15.8	0.04	3.4	3.6	0.12
ASR.in-domain	15.4	**14.0**	0.35	9.6	**9.0**	0.29
ASR.out-of-domain	58.2	**52.9**	0.10	42.8	**37.7**	0.23

Table 2: Black box regression accuracy. Lowest MAE is in bold font if statistically significant (p=0.05) according to the bootstrap resampling significance test (Koehn, 2004).

Pearson correlation	Segment-level (all)	Segment-level (within document)	Document-level
Black box regressor	0.23	0.09	0.44
Negative confidence	0.34	0.61	0.12

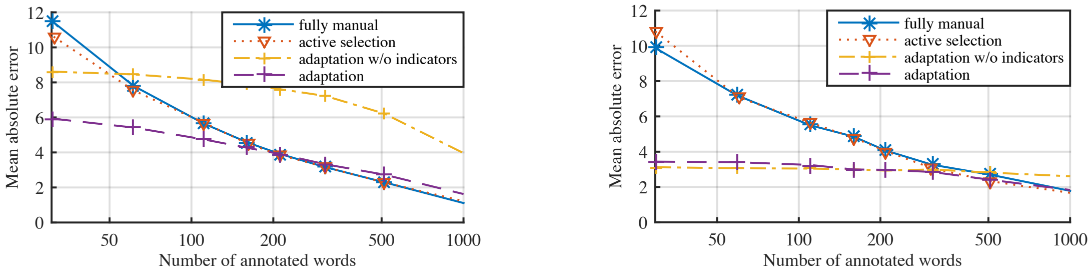
Table 3: Correlation for black box regressor vs. confidence scores with true WER labels on the ASR.conf-task dataset. Confidence scores are strong especially when evaluating the correlation on a per-document basis (averaging over documents), indicating that they may be more suitable for the active selection strategy than the black box approach.

of achieving positive segment correlation and supporting adaptation (§4.1). On document-level, the XT regressor outperforms the baseline in all but the MT.out-of-domain setting in terms of MAE, and correlation is stronger as well.

In Table 3 we also evaluate the performance of ASR confidence scores on the ASR.conf-task dataset. Since regressor outputs and confidences have different units, we compare them in terms of Pearson correlation. Compared to the black box regressor, it can be seen that confidence scores excel especially for the average within-document correlation. We would thus expect confidence scores to outperform black box regression for the active selection strategy.

5.2 MT Setting

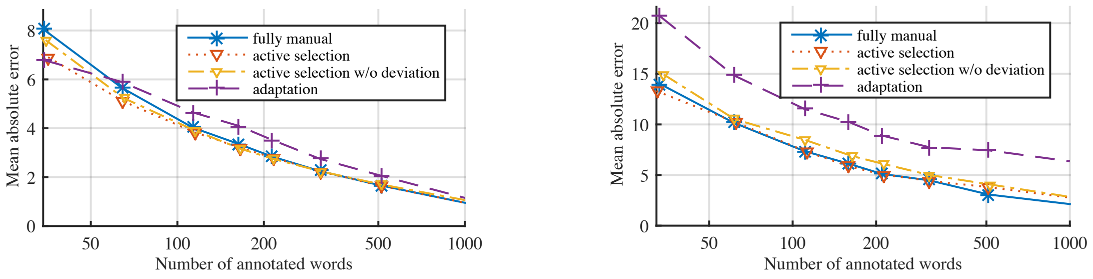
Figure 2 shows the results for the lightly supervised scenario for the MT datasets. Note that the fully automatic baseline corresponds to the leftmost point of the regressor adaptation curve. While the active selection strategy did not clearly outperform the fully manual baseline, regressor adaptation with enabled indicator features performed well. It can be seen that even a moderate amount of annotation improved the estimates, and the advantage over the fully manual baseline was especially strong for the out-of-domain setting. When removing the indicator features, regressor adaptation performs poorly in the in-domain setting. For the out-of-domain setting, performance changes only slightly. In both settings, it seems that the indicator features help to reach similar performance as the fully manual baseline for larger amounts of annotated data, suggesting that they help improve adaptation behavior.



(a) MT.in-domain

(b) MT.out-of-domain

Figure 2: Lightly supervised setting for MT.



(a) ASR.in-domain

(b) ASR.out-of-domain

Figure 3: Lightly supervised setting for ASR.

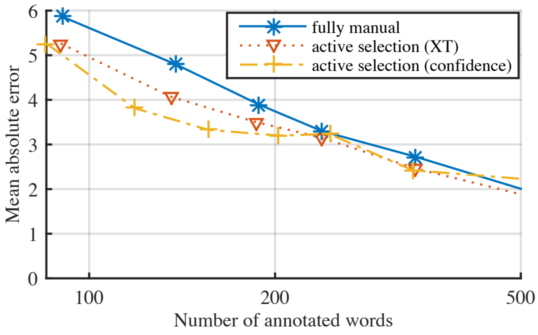


Figure 4: Active selection on the ASR.conf-task dataset, comparing the baseline and selection via XT regression scores and confidence scores.

5.3 ASR Setting

Figure 3 shows the results for the ASR datasets. The regressor adaptation strategy performs rather poorly. For active selection, we evaluate predicting deviations (§4.1), and observe small but consistent gains compared to the unaltered objective function. In result, we slightly outperform the fully manual baseline for the in-domain setting, and perform on par with the baseline for the out-of-domain setting.

However, the observed gains are probably too small to be considered worthwhile. We therefore also investigate replacing the XT regression scores by confidence scores provided directly by the ASR (§4.2). Here, XT regression was trained on data similar as in the test (in-domain setting), but again excluding all data from the particular document and system being tested. For the confidence scores, there is no distinction between in-domain and out-of-domain. Figure 4 reveals the more solid gains over both the fully manual baseline and active selection based on XT regression scores. We conclude that confidence scores are a promising way of reducing labeling effort in our slightly supervised quality estimation framework.

5.4 Discussion

As was seen in the above experiments, the proposed regressor adaptation strategy performed well for MT but not for ASR, whereas for the proposed active selection strategy it was the other way around. We also observe that for the MT datasets, there was relatively high between-segment variance compared to a relatively low between-document variance. This puts the fully manual baseline at a disadvantage, and may be the reason for the good regression-based results (fully automatic and regressor adaptation). In contrast, for the ASR datasets we observed relatively low between-segment variance and high between-document variance, which would put the annotation-based strategies at an advantage (fully manual and active selection). Scarton et al. (2015) made similar observations for MT, and we expect these findings to hold for other datasets. We also confirmed that results are similar when using BLEU+1 (Lin and Och, 2004) as the metric, instead of TER. Whether or not the situation changes when switching to very different metrics, such as non reference-based metrics, is left as a question for future work.

6 Relation to Prior Work

A good overview over the state-of-the-art in automatic QE for MT is given by Bojar et al. (2015) and Bojar et al. (2016), and for ASR by Ogawa et al. (2012) and Negri et al. (2014). Document-level QE was first explored by Soricut and Echihabi (2010) by exploiting document-level features, and was later improved by using sentence-level information (Soricut and Narsale, 2012; Specia et al., 2015; Bojar et al., 2015). Scarton and Specia (2014) and Scarton et al. (2015) argue that document-level quality metrics should consider discourse information that cannot be captured when processing segments individually, and explore features for QE that capture discourse information. Quantitative assessment of discourse information remains challenging (Bojar et al., 2016). Our aggregation approach supports only sentence-level information.

The out-of-domain case investigated in this work, i.e. predicting quality for a previously unknown system or task, has been found challenging in both ASR (Negri et al., 2014) and MT (de Souza et al., 2015a). The WMT 2015 shared task on QE considered only the scenario of training and testing on output produced by the same system (Bojar et al., 2015). The usual way to address domain mismatch when training data for all domains is available is via adaptation/multitask learning (Beck et al., 2014; de Souza et al., 2015b). Our indicator features can be seen as a form of multitask learning. A strategy for the case where no in-domain training data is available is to obtain such training data cheaply via active learning (Beck et al., 2013). This work is probably most similar in spirit to ours in that it attempts reliable quality estimation at low labeling costs.

A different line of research, crowd-sourced annotation, critically depends on quality control, as well. Approaches are usually based on comparing results between several workers (Passonneau and Carpenter, 2014), querying gold standard "testing" labels occasionally (Joglekar and Garcia-Molina, 2013), and/or automatically predicting quality (Roy et al., 2010; Gao et al., 2015). Our work can be seen as a generalization of the latter two, with the gold labels corresponding to our fully manual baseline, the automatic estimation corresponding to our fully automatic baseline, and the *workers* being our *systems*.

7 Conclusion

We proposed lightly supervised quality estimation at the document level, a framework that allows flexible quality estimation across changing domains and quality requirements, while requiring only moderate human annotation effort. We suggested two strategies for combining manual and automatic segment-level scores into a document-level estimate. The first strategy, regressor adaptation, was able to reduce annotation effort considerably for TER-based quality estimation in MT. The second strategy, active selection of segments for annotation, showed promising results for ASR quality estimation in terms of WER, when confidence scores are available.

As future work, we suggest exploring further evaluation metrics, in particular non reference-based metrics. Depending on the metrics, user interfaces that allow manual annotation at low effort should be designed. Also, the confidence-based active selection strategy may be worth investigating for MT.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful suggestions. The project leading to this application has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement n° 645452.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR : An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine translation and/or Summarization*, pages 65–72.

Daniel Beck, Lucia Specia, and Trevor Cohn. 2013. Reducing Annotation Effort for Quality Estimation via Active Learning. In *Association for Computational Linguistics Conference (ACL)*, pages 543–548, Sofia, Bulgaria.

Daniel Beck, Kashif Shah, and Lucia Specia. 2014. SHEF-Lite 2.0: Sparse Multi-task Gaussian Processes for Translation Quality Estimation. In *Association for Computational Linguistics Conference (ACL)*, pages 307–312, Baltimore, USA.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Workshop on Statistical Machine Translation (WMT)*, pages 1–46, Lisbon, Portugal.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Conference on Machine Translation (WMT)*, pages 131–198.

Hal Daumé III. 2007. Frustratingly Easy Domain Adaptation. In *Association for Computational Linguistic (ACL)*, pages 256–263, Prague, Czech Republic.

José G. C. de Souza, Matteo Negri, Elisa Ricci, and Marco Turchi. 2015a. Online Multitask Learning for Machine Translation Quality Estimation. In *Association for Computational Linguistics (ACL)*, pages 219–228, Beijing, China.

José G. C. de Souza, Hamed Zamani, Matteo Negri, Marco Turchi, and Daniele Falavigna. 2015b. Multitask Learning for Adaptive Quality Estimation of Automatically Transcribed Utterances. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies Conference (NAACL-HLT)*, pages 714–724, Denver, USA.

Mingkun Gao, Wei Xu, and Chris Callison-Burch. 2015. Cost Optimization for Crowdsourcing Translation. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies Conference (NAACL-HLT)*, pages 705–713, Denver, USA.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Yvette Graham. 2015. Improving Evaluation of Machine Translation Quality Estimation. In *Association for Computational Linguistics (ACL)*, pages 1804–1813, Beijing, China.

Kim Harris, Aljoscha Burchardt, Georg Rehm, and Lucia Specia. 2016. Technology Landscape for Quality Evaluation: Combining the Needs of Research and Industry. In *LREC Workshop on Translation Evaluation*, pages 50–54, Portorož, Slovenia.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer, second edition.

Manas Joglekar and Hector Garcia-Molina. 2013. Evaluating the Crowd with Confidence. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 686–694, Chicago, USA.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395, Barcelona, Spain.

Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: A method for evaluating automatic evaluation metrics for machine translation. In *International Conference on Computational Linguistics (COLING)*, pages 501–507, Geneva, Switzerland.

Matteo Negri, Marco Turchi, G. C. de Souza, and Daniele Falavigna. 2014. Quality Estimation for Automatic Speech Recognition. In *International Conference on Computational Linguistics (COLING)*, pages 1813–1823, Dublin, Ireland.

Atsunori Ogawa, Takaaki Hori, and Atsushi Nakamura. 2012. Error Type Classification and Word Accuracy Estimation using Alignment Features from Word Confusion Network. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4925–4928, Kyoto, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Association for Computational Linguistic (ACL)*, pages 311–318, Philadephia, Pennsylvania.

Rebecca J. Passonneau and Bob Carpenter. 2014. The Benefits of a Model of Annotation. *Transactions of the Association for Computational Linguistics (TACL)*, 2:311–326.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Brandon C. Roy, Soroush Vosoughi, and Deb Roy. 2010. Automatic Estimation of Transcription Accuracy and Difficulty. In *Annual Conference of the International Speech Communication Association (InterSpeech)*, Makuhari, Japan.

Carolina Scarton and Lucia Specia. 2014. Document-level translation quality estimation: exploring discourse and pseudo-references. In *European Association for Machine Translation (EAMT)*, pages 101 – 108.

Carolina Scarton, Marcos Zampieri, Mihaela Vela, Josef van Genabith, and Lucia Specia. 2015. Searching for Context: a Study on Document-Level Labels for Translation Quality Estimation. In *Conference of the European Association for Machine Translation (EAMT)*, pages 121–128, Antalya, Turkey.

Radu Soricut and Abdessamad Echihabi. 2010. TrustRank: Inducing trust in automatic translations via ranking. In *Association for Computational Linguistic (ACL)*, pages 612–621.

Radu Soricut and Sushant Narsale. 2012. Combining quality prediction and system selection for improved automatic translation output. In *Association for Computational Linguistic (ACL)*, pages 163–170.

Lucia Specia, Kashif Shah, Jose G. C. de Souza, and Trevor Cohn. 2013. QuEst - A Translation Quality Estimation Framework. In *Association for Computational Linguistics: System Demonstrations*, pages 79–84, Sofia, Bulgaria.

Lucia Specia, Gustavo H. Paetzold, and Carolina Scarton. 2015. Multi-level Translation Quality Prediction with QUEST++. In *Association of Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 115–120, Beijing, China.