MT Quality Estimation: The CMU System for WMT'13

Almut Silja Hildebrand
Carnegie Mellon University
Pittsburgh, USA
silja@cs.cmu.edu

Stephan Vogel
Qatar Computing Research Institute
Doha, Qatar
svogel@qf.org.qa

Abstract

In this paper we present our entry to the WMT'13 shared task: Quality Estimation (QE) for machine translation (MT). We participated in the 1.1, 1.2 and 1.3 sub-tasks with our QE system trained on features from diverse information sources like MT decoder features, n-best lists, mono- and bi-lingual corpora and giza training models. Our system shows competitive results in the workshop shared task.

1 Introduction

As MT becomes more and more reliable, more people are inclined to use automatically translated texts. If coming across a passage that is obviously a mistranslation, any reader would probably start to doubt the reliability of the information in the whole article, even though the rest might be adequately translated. If the MT system had a QE component to mark translations as reliable or possibly erroneous, the reader would know to use information from passages marked as bad translations with caution, while still being able to trust other passages. In post editing a human translator could use translation quality annotation as an indication to whether editing the MT output or translating from scratch might be faster. Or he could use this information to decide where to start in order to improve the worst passages first or skip acceptable passages altogether in order to save time. Confidence scores can also be useful for applications such as cross lingual information retrieval or question answering. Translation quality could be a valuable ranking feature there.

Most previous work in the field estimates confidence on the sentence level (e.g. Quirk et al. (2004)), some operate on the word level (e.g. Ueffing and Ney (2007), Sanchis et al. (2007), and Bach et al. (2011)), whereas Soricut and Echihabi (2010) use the document level.

Various classifiers and regression models have been used in QE in the past. Gandrabur and Foster (2003) compare single layer to Multi Layer Perceptron (MLP), Quirk et al. (2004) report that Linear Regression (LR) produced the best results in a comparison of LR, MLP and SVM, Gamon et al. (2005) use SVM, Soricut and Echihabi (2010) find the M5P tree works best among a number of regression models, while Bach et al. (2011) define the problem as a word sequence labeling task and use MIRA.

The QE shared task was added to the WMT evaluation campaign in 2012 (Callison-Burch et al., 2012), providing standard training and test data for system development.

2 WMT'13 Shared Task

In this WMT Shared Task for Quality Estimation[1] there were tasks for sentence and word level QE. We participated in all sub-tasks for Task 1: Sentence-level QE.

Task 1.1: Scoring and ranking for post-editing effort focuses on predicting HTER per segment for the translations of one specific MT system. Task 1.2: System selection/ranking required to predict a ranking for up to five translations of the same source sentence by different MT systems. The training data provided manual labels for ranking including ties. Task 1.3: Predicting post-editing time participants are asked to predict the time in seconds a professional translator will take to post edit each segment.

[1] http://www.statmt.org/wmt13/quality-estimation-task.html

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 373–379,
Sofia, Bulgaria, August 8-9, 2013

Besides the training data with labels, for each of these tasks additional resources were provided. These include bilingual training corpora, language models, 1000-best lists, models from giza and moses training and various other statistics and models depending on task and language pair.

3 Features

3.1 Language Models

To calculate language model (LM) features, we train traditional n-gram language models with n-gram lengths of four and five using the SRILM toolkit (Stolcke, 2002). We calculate our features using the KenLM toolkit (Heafield, 2011). We normalize all our features with the target sentence length to get an average word feature score, which is comparable for translation hypotheses of different length. In addition to the LM probability we record the average n-gram length found in the language model for the sentence, the total number of LM OOVs and OOVs per word, as well as the maximum and the minimum word probability of the sentence, six features total.

We use language models trained on source language data and target language data to measure source sentence difficulty as well as translation fluency.

3.2 Distortion Model

The moses decoder uses one feature from a distance based reordering model and six features from a lexicalized reordering model: Given a phrase pair, this model considers three events Monotone, Swap, and Discontinuous in two directions Left and Right. This results in six events: LM (left-monotone), LS (left-swap), LD (left-discontinuous) and RM (right-monotone), RS, RD.

These distortion features are calculated for each phrase. For a total sentence score we normalize by the phrase count for each of the seven features.

3.3 Phrase Table

From the phrase table we use the features from the moses decoder output: inverse phrase translation probability, inverse lexical weighting, direct phrase translation probability and direct lexical weighting. For a total sentence score we normalize by the phrase count. We use the number of phrases used to generate the hypothesis and the average phrase length as additional features, six features total.

3.4 Statistical Word Lexica

From giza training we use IBM-4 statistical word lexica in both directions. We use six probability based features as described in Hildebrand and Vogel (2008): Normalized probability, maximum word probability and word deletion count from each language direction.

To judge the translation difficulty of each word in the source sentence we collect the number of lexicon entries for each word similar to Gandrabur and Foster (2003). The intuition is, that a word with many translation alternatives in the word-to-word lexicon is difficult to translate while a word with only a few translation choices is easy to translate.

In fact it is not quite this straight forward. There are words in the lexicon, which have many lexicon entries, but the probability for them is not very equally distributed. One entry has a very high probability while all others have a very low one - not much ambiguity there. Other words on the other hand have several senses in one language and therefore are translated frequently into two or three different words in the target language. There the top entries in the lexicon might each have about 30% probability. To capture this behavior we do not only count the total number of entries but also the number of entries with a probability over a threshold of 0.01.

For example one word with a rather high number of different translations in the English-Spanish statistical lexicon is the period (.) with 1570 entries. It has only one translation with a probability over the threshold which is the period (.) in Spanish at a probability of 0.9768. This shows a clear choice and rather little ambiguity despite the high number of different translations in the lexicon.

For each word we collect the number of lexicon entries, the number of lexicon entries over the threshold, the highest probability from the lexicon and whether or not the word is OOV. If a word has no lexicon entry with a probability over the threshold we exclude the word from the lexicon for this purpose and count it as an OOV. As sentence level features we use the sum of the word level features normalized by the sentence length as well as the total OOV count for the sentence, which results in five features.

3.5 Sentence Length Features

The translation difficulty of a source sentence is often closely related to the sentence length, as longer sentences tend to have a more complex structure. Also a skewed ratio between the length of the source sentence and its translation can be an indicator for a bad translation.

We use plain sentence length features, namely the source sentence length, the translation hypothesis length and their ratio as introduced in Quirk (2004).

Similar to Blatz et al. (2004) we use the n-best list as an information source. We calculate the average hypothesis length in the n-best list for one source sentence. Then we compare the current hypothesis to that and calculate both the diversion from that average as well as their ratio. We also calculate the source-target ratio to this average hypothesis length.

To get a representative information on the length relationship of translations from the source and target languages in question, we use the parallel training corpus. We calculate the overall language pair source to target sentence length ratio and record the diversion of the current hypothesis' source-target ratio from that.

The way sentences are translated from one language to another might differ depending on how complex the information is, that needs to be conveyed, which in turn might be related to the sentence length and the ratio between source and translation. As a simple way of capturing this phenomenon we divide the parallel training corpus into three classes (short, medium, long) by the length of the source language sentence. The boundaries of these classes are the mean 26.84 plus and minus the standard deviation 14.54 of the source sentence lengths seen in the parallel corpus. We calculate the source/target length ratio for each of the three classes separately. The resulting statistics for the parallel training corpora can be found in Table 1. For English - Spanish the ratio for all classes is close to one, for other language pairs these differ more clearly.

As features for each hypothesis we use a binary indicator for its membership to each class and its deviation from the length ratio of its class. This results in 12 sentence length related features in total.

	En train
number of sentences	1,714,385
average length	26.84
standard deviation	14.54
class short	0 - 12.29
class medium	12.29 - 41.38
class long	41.38 - 100
s/t ratio overall	0.9624
s/t ratio for short	0.9315
s/t ratio for medium	0.9559
s/t ratio for long	0.9817

Table 1: Sentence Length Statistics for the English-Spanish Parallel Corpus

3.6 Source Language Word and Bi-gram Frequency Features

The length of words is often related to whether they are content words and how frequently they are used in the language. Therefore we use the maximum and average word length as features.

Similar to Blatz et al. (2004) we sort the vocabulary of the source side of the training corpus by occurrence frequency and then divide it into four parts, each of which covers 25% of all tokens. As features we use the percentage of words in the source sentence that fall in each quartile. Additionally we use the number and percentage of source words in the source sentence that are OOV or very low frequency, using count 2 as threshold. We also collect all bigram statistics for the corpus and calculate the corresponding features for the source sentence based on bigrams. This adds up to fourteen features from source word and corpus statistics.

3.7 N-Best List Agreement & Diversity

We use the three types of n-best list based features described in Hildebrand and Vogel (2008): Position Dependent N-best List Word Agreement, Position independent N-best List N-gram Agreement and N-best List N-gram Probability.

To measure n-best list diversity, we compare the top hypothesis to the 5th, 10th, 100th, 200th, 300th, 400th and 500th entry in the n-best list (where they exist) to see how much the translation changes throughout the n-best list. We calculate the Levenshtein distance (Levenshtein, 1966) between the top hypothesis and the three lower ranked ones and normalize by the sentence length

of the first hypothesis. We also record the n-best list size and the size of the vocabulary in the n-best list for each source sentence normalized by the source sentence length.

Fifteen agreement based and nine diversity based features add up to 24 n-best list based features.

3.8 Source Parse Features

The intuition is that a sentence is harder to translate, if its structure is more complicated. A simple indicator for a more complex sentence structure is the presence of subclauses and also the length of any clauses and subclauses. To obtain the clause structure, we parse the source language sentence using the Stanford Parser[2] (Klein and Manning, 2003). Features are: The number of clauses and subclauses, the average clause length, and the number of sentence fragments found. If the parse does not contain a clause tag, it is treated as one clause which is a fragment.

3.9 Source-Target Word Alignment Features

A forced alignment algorithm utilizes the trained alignment models from the MT systems GIZA (Och and Ney, 2003) training to align each source sentence to each translation hypothesis.

We use the score given by the word alignment models, the number of unaligned words and the number of NULL aligned words, all normalized by the sentence length, as three separate features. We calculate those for both language directions. Hildebrand and Vogel (2010) successfully applied these features in n-best list re-ranking.

3.10 Cohesion Penalty

Following the cohesion constraints described in Bach et al. (2009) we calculate a cohesion penalty for the translation based on the dependency parse structure of the source sentence and the word alignment to the translation hypothesis. To obtain these we use the Stanford dependency parser (de Marneffe et al., 2006) and the forced alignment from Section 3.9.

For each head word we collect all dependent words and also their dependents to form each complete sub-tree. Then we project each sub-tree onto the translation hypothesis using the alignment. We test for each sub-tree, whether all projected words in the translation are next to each other (cohesive) or if there are gaps. From the collected gaps we subtract any unaligned words. Then we count the number of gaps as cohesion violations as well as how many words are in each gap. We go recursively up the tree, always including all sub-trees for each head word. If there was a violation in one of the sub-trees it might be resolved by adding in its siblings, but if the violation persists, it is counted again.

4 Classifiers

For all experiments we used the Weka[3] data mining toolkit described in Hall et. al. (2009) to compare four different classifiers: Linear Regression (LR), M5P tree (M5Ptree), Multi Layer Perceptron (MLP) and Support Vector Machine for Regression (SVM). Each of these has been identified as effective in previous publications. All but one of the Weka default settings proved reliable, changing the learning rate for the MLP from default: 0.3 to 0.01 improved the performance considerably. We report Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for all results.

5 Experiment Results

For Tasks 1.1 and 1.3 we used the 1000-best output provided. As first step we removed duplicate entries in these n-best list. This brought the size down to an average of 152.9 hypotheses per source sentence for the Task 1.1 training data, 172.7 on the WMT12 tests set and 204.3 hypotheses per source sentence on the WMT13 blind test data. The training data for task 1.3 has on average 129.0 hypothesis per source sentence, the WMT13 blind test data 129.8.

In addition to our own features described above we extracted the 17 features used in the WMT12 baseline for all sub-tasks via the software provided for the WMT12-QE shared task.

5.1 Task 1.1

Task 1.1 is to give a quality score between 0 and 1 for each segment in the test set, predicting the HTER score for the segment and also to give a rank for each segment, sorting the entire test set from best quality of translation to worst.

For Task 1.1 our main focus was the scoring task. We did submit a ranking for the blind test

[2] http://nlp.stanford.edu/software/lex-parser.shtml

[3] http://www.cs.waikato.ac.nz/ml/weka/

wmt12-test: WMT12 manual quality labels					
WMT12 best system: Language Weaver			0.61 - 0.75		
WMT12 baseline system			0.69 - 0.82		
feat. set	#feat	LR	M5Pt	MLP	SVM
full	117	0.617 - 0.755	0.618 - 0.756	0.619 - 0.773	0.609 - 0.750
no WMT12-base	100	0.618 - 0.766	0.618 - 0.767	0.603 - 0.757	0.611 - 0.761
slim	69	0.621 - 0.767	0.621 - 0.766	0.614 - 0.768	0.627 - 0.773
wmt12-test: HTER					
full	117	0.125 - 0.162	0.126 - 0.163	**0.122 - 0.156**	0.121 - 0.156
no WMT12-base	100	0.124 - 0.160	0.123 - 0.159	0.125 - 0.159	**0.121 - 0.155**
slim	69	0.125 - 0.161	0.126 - 0.161	0.124 - 0.159	0.123 - 0.158
wmt13-test: HTER					
WMT12 baseline system			0.148 - 0.182		
full	117	0.146 - 0.183	0.147 - 0.185	0.156 - 0.199	0.142 - 0.180
no WMT12-base	100	0.144 - 0.180	0.144 - 0.180	0.156 - 0.203	0.139 - 0.176
slim	69	0.147 - 0.182	0.147 - 0.181	0.153 - 0.194	0.142 - 0.177

Table 2: Task 1.1: Results in MAE and RMSE on the WMT12 test set for WMT12 manual labels as well as WMT13 HTER as target class and the WMT13 test set for HTER

set as well, which resulted from simply sorting the test set by the estimated HTER per segment.

In Table 2 we show the results for some experiments comparing the performance of different feature sets and classifiers. For development we used the WMT12-QE test set and both the WMT12 manual labels as well as HTER as target class. We compared the impact of removing the 17 WMT12-baseline features "no WMT12-base" and training a "slim" system by removing nearly half the features, which showed to have a smaller impact on the overall performance in preliminary experiments. Among the removed features are n-best list based features, redundant features between ours, the moses based and the base17 features and some less reliable features like e.g. the lexicon deletion features, who's thresholds need to be calibrated carefully for each new language pair. We submitted the full+MLP and the no-WMT12-base+SVM output to the shared task, shown in bold in the table.

The official result for our system for task 1.1 on the WMT13 blind data is MAE 13.84, RMSE 17.46 for the no-WMT12-base+SVM system and MAE 15.25 RMSE 18.97 for the full+MLP system. Surprising here is the fact that our full system clearly outperforms the 17-feature baseline on the WMT12 test set, but is behind it on the WMT13 blind test set. (Baseline bb17 SVM: MAE 14.81, RMSE 18.22) Looking at the WMT13 test set results, we should have chosen the slim+SVM system variant.

5.2 Task 1.2

Task 1.2 asks to rank different MT systems by translation quality on a segment by segment basis.

Since the manually annotated ranks in task 1.2 allowed ties, we treated them as quality scores and ran the same QE system on this data as we did for task 1.1. We submitted the full-MLP output with the only difference that for this data set the decoder based features were not available. We rounded the predicted ranks to integer. Since the training data contains many ties we did not employ a strategy to resolve ties.

As a contrastive approach we ran the hypothesis selection system described in Hildebrand and Vogel (2010) using the BLEU MT metric as ranking criteria. For this system it would have been very beneficial to have access to the n-best lists for the different system's translations. The BLEU score for the translation listed as the first system for each source sentence would be 30.34 on the entire training data. We ran n-best list re-ranking using MERT (Och, 2003) for two feature sets: The full feature set, 100 features in total and a slim feature set with 59 features. For the slim feature set we removed all features that are solely based on

the source sentence, since those have no impact on re-ranking an n-best list. The BLEU score for the training set improved to 45.25 for the full feature set and to 45.76 for the slim system. Therefore we submitted the output of the slim system to the shared task. This system does not predict ranks directly, but estimates ranking according to BLEU gain on the test set. Therefore the new ranking is always ranks 1-5 without ties.

The official result uses Kendalls tau with and without ties penalized. Our two submissions score: -0.11 / -0.11 for the BLEU optimized system and -0.63 / 0.23 for the classifier system. The classifier system is the best submission in the "ties ignored" category.

5.3 Task 1.3

Task 1.3 is to estimate post editing time on a per segment basis.

In absence of a development test set we used 10-fold cross-validation on the training data to determine the best feature set and classifier for the two submissions. Table 3 shows the results on our preliminary tests for four classifiers and three feature sets. The "no pr." differs from the full feature set only by removing the provided features, in this case the 17 WMT12-baseline features and the "translator ID" and "nth in doc" features. For the "slim" system run the feature set size was cut in half in order to prevent overfitting to the training data since the training data set is relatively small. We used the same criteria as in Task 1.1. For the shared task we submitted the full+SVM and slim+LR variants, shown in bold in the table.

The official result for our entries on the WMT13 blind set in MAE and RMSE are: 53.59 - 92.21 for the full system and 51.59 - 84.75 for the slim system. The slim system ranks 3rd for both metrics and outperforms the baseline at 51.93 - 93.36.

6 Conclusions

In this WMT'13 QE shared task we submitted to the 1.1, 1.2 and 1.3 sub-tasks. In development we focused on the scoring type tasks.

In general there don't seem to be significant differences between the different classifiers.

Surprising is the fact that our full system for task 1.1 clearly outperforms the 17-feature baseline on the WMT12 test set, but is behind it on the WMT13 blind test set. This calls into question whether the performance on the WMT12 test set was the right criterium for selecting a system variant for submission.

The relative success of the "slim" system variant over the full feature set shows that our system would most likely benefit from a sophisticated feature selection method. We plan to explore this in future work.

References

Nguyen Bach, Stephan Vogel, and Colin Cherry. 2009. Cohesive constraints in a beam search phrase-based decoder. In *HLT-NAACL (Short Papers)*, pages 1–4.

Nguyen Bach, Fei Huang, and Yaser Al-Onaizan. 2011. Goodness: A method for measuring machine translation confidence. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 211–219, Portland, Oregon, USA, June. Association for Computational Linguistics.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. Technical report, Final report JHU / CLSP 2003 Summer Workshop, Baltimore.

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.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC-06*.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level mt evaluation without reference translations: Beyond language modeling. In *In European Association for Machine Translation (EAMT*.

Simona Gandrabur and George Foster. 2003. Confidence estimation for translation prediction. In *In Proceedings of CoNLL-2003*, page 102.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July. Association for Computational Linguistics.

feat. set	#feat	class.	10-fold cross	train	WMT13 test
full	119	LR	45.73 - 73.52	39.74 - 63.92	54.45 - 88.68
full	119	M5Pt	44.49 - 74.05	35.81 - 57.36	50.05 - 85.22
full	119	MLP	48.05 - 75.68	41.03 - 68.70	54.38 - 88.93
full	119	SVM	**40.88 - 73.61**	34.70 - 69.69	53.74 - 92.26
no pr	100	LR	46.06 - 74.94	40.39 - 66.00	52.13 - 86.68
no pr	100	M5Pt	43.80 - 74.30	36.80 - 59.47	50.86 - 87.42
no pr	100	MLP	47.70 - 75.41	39.85 - 68.30	52.39 - 87.93
no pr	100	SVM	41.35 - 74.68	35.59 - 70.99	52.87 - 92.22
slim	59	LR	**44.72 - 73.86**	41.14 - 67.44	51.71 - 84.83
slim	59	M5Pt	43.77 - 74.43	35.26 - 56.84	57.75 - 102.68
slim	59	MLP	46.98 - 74.38	40.35 - 69.79	51.06 - 85.48
slim	59	SVM	40.42 - 74.47	36.88 - 71.59	51.09 - 90.18

Table 3: Task 1.3: Results in MAE and RMSE for 10-fold cross validation and the whole training set as well as the WMT13 blind test set

Almut Silja Hildebrand and Stephan Vogel. 2008. Combination of machine translation systems via hypothesis selection from combined n-best lists. In *MT at work: Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 254–261, Waikiki, Hawaii, October. Association for Machine Translation in the Americas.

Almut Silja Hildebrand and Stephan Vogel. 2010. CMU system combination via hypothesis selection for WMT'10. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 307–310. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

Christopher B. Quirk. 2004. Training a sentence-level machine translation confidence measure. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, pages 825–828, Lisbon, Portugal, May. LREC.

Alberto Sanchis, Alfons Juan, Enrique Vidal, and Departament De Sistemes Informtics. 2007. Estimation of confidence measures for machine translation. In *In Procedings of Machine Translation Summit XI*.

Radu Soricut and Abdessamad Echihabi. 2010. Trustrank: Inducing trust in automatic translations via ranking. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 612–621, Uppsala, Sweden, July. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings International Conference for Spoken Language Processing*, Denver, Colorado, September.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.