Quality Estimation for Machine Translation output using linguistic analysis and decoding features

Eleftherios Avramidis
German Research Center for Artificial Intelligence (DFKI)
Berlin, Germany
eleftherios.avramidis@dfki.de

Abstract

We describe a submission to the WMT12 Quality Estimation task, including an extensive Machine Learning experimentation. Data were augmented with features from linguistic analysis and statistical features from the SMT search graph. Several Feature Selection algorithms were employed. The Quality Estimation problem was addressed both as a regression task and as a discretised classification task, but the latter did not generalise well on the unseen testset. The most successful regression methods had an RMSE of 0.86 and were trained with a feature set given by Correlation-based Feature Selection. Indications that RMSE is not always sufficient for measuring performance were observed.

1 Introduction

As Machine Translation (MT) gradually gains a position into production environments, the need for estimating the quality of its output is increasing. Various use cases refer to it as input assessment for Human Post-editing, as an extension for Hybrid MT or System Combination, or even a method for improving components of existing MT systems.

With the current submission we are trying to address the problem of assigning a quality score to a single MT output per source sentence. Previous work includes regression methods for indicating a binary value of correctness (Quirk, 2001; Blatz et al., 2004; Ueffing and Ney, 2007), human-likeness (Gamon et al., 2005) or continuous scores (Specia et al., 2009). As we also work with continuous scores, we are making an effort to combine previous feature acquisition sources, such as language modelling (Raybaud et al., 2009), language fluency checking (Parton et al., 2011), parsing (Sánchez-Martinez, 2011; Avramidis et al., 2011) and decoding statistics (Specia et al., 2009; Avramidis, 2011). The current submission combines such previous observations in a combinatory experimentation on feature sets, feature selection methods and Machine Learning (ML) algorithms.

The structure of the submission is as follows: The approach is defined and the methods are described in section 2, including features acquisition, feature selection and learning. Section 3 includes information about the experiment setup whereas the results are discussed in Section 4.

2 Methods

2.1 Data and basic approach

This contribution has been built based on the data released for the Quality Estimation task of the Workshop on Machine Translation (WMT) 2012 (Callison-Burch et al., 2012). The organizers provided an English-to-Spanish development set and a test set of 1832 and 422 sentences respectively, derived from WMT09 and WMT10 datasets. For each source sentence of the development set, participants were offered one translation generated by a state-of-the-art phrase-based SMT system. The quality of each SMT translation was assessed by human evaluators, who provided a quality score in the range 1-5. Additionally, statistics and processing information from the execution of the SMT decoding algorithm were given.

The approach presented here is making use of the source sentences, the SMT output and the quality scores in order to follow a typical ML paradigm:

Proceedings of the 7th Workshop on Statistical Machine Translation, pages 84–90,
Montréal, Canada, June 7-8, 2012. ©2012 Association for Computational Linguistics

sentence	suggestion
...los líderes de la Unión han descrito como **deducciones político** ...	*number agreement*
La articular **y ideológicamente** convencido de asesino de masas ...	*transform "y" to "e"*
Right after **hearing** about it, he described it as a "challenge..."	*disambiguate -ing*

Table 1: Sample suggestions generated by rule-based language checking tools, observed in development data

each source and target sentence of the development set are being analyzed to generate a feature vector. One training sample is formed out of the feature vector and the quality score (i.e. as a class value) of each sentence. A ML algorithm is consequently used to train a model given the training samples. The performance of each model is evaluated upon a part of the development set that was kept-out from training.

2.2 Acquiring Features

The features were obtained from two sources: the decoding process and the analysis of the text of the source and the target sentence. The two steps are explained below.

2.2.1 Features from text analysis

The following features were generated with the use of tools for the statistical and/or linguistic analysis of the text. The baseline features included:

- **Tokens count**: Count of tokens in the source and the translated sentence and their ratio, unknown words and also occurrences of the target word within the translated sentence (averaged for all words in the hypothesis - type/token ratio)
- **IBM1-model lookup**: Average number of translations per source word in the sentence, unweighted or weighted by the inverse frequency of each word in the source corpus
- **Language modeling**: Language model probability of the source and translated sentence
- **Corpus lookup**: percentage of unigrams / bigrams / trigrams in quartiles 1 and 4 of frequency (lower and higher frequency words) in a corpus of the source language

Additionally, the following linguistically motivated features were also included:

- **Parsing**: PCFG Parse (Petrov et al., 2006) log-likelihood, size of n-best tree list, confidence for the best parse, average confidence of all parse trees. Ratios of the mentioned target features to the corresponding source features.
- **Shallow grammatical match**: The number of occurences of particular node tags on both the source and the target was counted on the PCFG parses. Additionally, the ratio of the occurences of each tag in the target sentence by the corresponding occurences on the source sentence.
- **Language quality check**: Source and target sentences were subject to automatic rule-based *language quality checking*, providing a wide range of quality suggestions concerning **style**, **grammar** and **terminology**, summed up in an overall quality score. The process employed 786 rules for English and 70 rules for Spanish. We counted the occurences of every rule match in each sentence and the number of characters it affected. Sample rule suggestions can be seen in Table 1.

2.2.2 Features from the decoding process

The organisers provided a verbose output of the decoding process, including probabilistic scores from all steps of the execution of the translation search. We added the scores appearing once per sentence (i.e. referring to the best hypothesis), whereas for the ones being modified over the generation graph, their average (avg), variance (var) and standard deviation (std) was calculated. These features are:

- the log of the phrase translation probability (pC) and the phrase future cost estimate (c)
- the score component vector including the distortion scores ($d_{1...7}$), word penalty, translation scores (e.g. a_1: inverse phrase translation probability, a_2: inverse lexical weighting)

2.3 Feature Selection

Experience has shown difficulties in including hundreds of features into training a statistical model. Several algorithms (such as Naïve Bayes) require statistically-independent features. For others, a search space of hundreds of features may impose increased computational complexity, which is often unsustainable in the time and resources allocated. In these cases we therefore applied several common *Feature Selection* approaches, in order to reduce the available features to an affordable number.

We used the Feature Selection algorithms of *Relieff* (Kononenko, 1994), *Information Gain* and *Gain Ratio* (Kullback and Leibler, 1951), and *Correlation-based Feature Selection* (Hall, 2000). The latter is known for producing feature sets highly correlated with the class, yet uncorrelated with each other; selection was done in two variations, *greedy stepwise* and *best first*.

The data were discretised according to the algorithm requirements and features were scored in a 10-fold cross-validation.

2.4 Machine Learning

We tried to approach the issue with two distinct modelling approaches, *classification* and *regression*.

2.4.1 Classification algorithms

In an effort to interpret Quality Estimation as a classification problem, we expect to build models that are able to assign a discrete value, as a measure of sentence quality. This bears some relation to the way the quality scores were generated; humans were asked to provide an (integer) quality score in the range 1-5. In our case, we try to build classifiers that do the same, but are also able to assign values with smaller intervals. For this purpose, we set up 4 sub-experiments, where the class value in our data was rounded up to intervals of 0.25, 0.5, 0.7 and 1.0 respectively.

In this part of the experiment we used the *Naïve Bayes*, *k-nearest-neighbours* (kNN), *Support Vector Machines* (SVM) and *Tree classification* algorithms. Naïve Bayes' probabilities for our continuous features were estimated with *locally weighted linear regression* (Cleveland, 1979).

2.4.2 Regression algorithms

Regression algorithms produce a model for directly predicting a quality score with continuous values. Experimentation here included *Partial Least Squares Regression* (Stone and Brooks, 1990), *Multivariate Adaptive Regression Splines – MARS* (Friedman, 1991), *Lasso* (Tibshirani, 1994) and *Linear Regression*.

3 Experiment and Results

3.1 Implementation

PCFG parsing features were generated on the output of the Berkeley Parser (Petrov and Klein, 2007), trained over an English and a Spanish treebank (Mariona Taulé and Recasens, 2008). N-gram features have been generated with the SRILM toolkit (Stolcke, 2002). The *Acrolinx IQ*[1] was used to parse the source side, whereas the *Language Tool*[2] was applied on both sides.

The feature selection and learning algorithms were implemented with the Orange (Demšar et al., 2004) and Weka (Hall et al., 2009) toolkits.

3.2 Experiment structure

The methods explained in the previous section provide a wide range of experiment parameters. Consequently, we tried to extensively test all the possible parameter combinations. The development data were separated in two sets, one "training" set and one "keep-out" set, used to test the predictions. In order to give learners better coverage over the data, the development set was split in two ways (70% training - 30% test and 90% training - 10% test), so that all experiments get performed under both settings. The scores of these two were averaged[3].

3.3 Results

The small size of the dataset allowed for fast training and testing of the discrete classification problem, where we could execute 370 experiments. The regression problem was considerably slower, as only 36 experiments concluded in time.

[1] http://www.acrolinx.com (proprietary)

[2] http://languagetool.org (open-source)

[3] Given the disparity of the test sizes, it would have in principle been better to use a weighted average. Though, this would not have lead to significant differences in the results.

			5-fold		avg 70-30%, 90-10% folds			
algorithm	feat. set	discr.	CA	AUC	RMSE	MAE	interval	
Tree	#17, #20	0.25	15.40	54.10	0.84	0.67	1.5	5.0
Tree	#23	0.25	14.60	53.50	0.85	0.68	2.0	5.0
Tree	#12	0.25	13.90	52.00	0.86	0.69	1.8	5.0
Tree	#4	0.25	14.50	53.70	0.86	0.69	2.0	5.0
SVM	#16	0.25	16.00	60.40	0.86	0.69	3.2	3.2
kNN	#22	0.25	12.30	55.50	1.00	0.78	2.0	5.0
Tree	#21	0.50	22.70	54.60	0.87	0.69	2.0	5.0
SVM	#19	0.50	22.40	60.20	0.91	0.73	2.8	5.0
kNN	#12	0.50	20.00	54.70	0.98	0.78	2.2	5.0
Naive	#6	0.50	21.20	59.40	0.99	0.76	1.2	5.0
Tree	#9	0.70	32.70	53.30	0.89	0.71	3.5	4.9
kNN	#12	0.70	28.20	56.10	0.93	0.73	2.5	4.9
SVM	#18	0.70	30.90	55.60	0.97	0.77	3.5	4.2
Tree	#22	1.00	40.30	55.70	0.90	0.71	2.0	5.0
kNN	#22	1.00	40.90	59.10	0.96	0.76	2.5	5.0
Naive	#23	1.00	41.00	65.50	1.02	0.78	1.2	5.0
SVM	#6	1.00	36.60	51.10	1.02	0.84	3.0	4.0

Table 2: Indicative discretised classification results, sorted by best performance and discretisation interval. Classification Accuracy (AC), Area Under Curve (AUC), Root Mean Square Error (RMSE) and Mean Average Error (MAE), Largest Error Percentage (LEP) and Smallest Error Percentage (SEP)

Feature generation resulted (described in Section 2.2) into 266 features, while 90 of them derived from language checking. Feature selection suggested several feature sets containing between 30 and 80 features. We ended up defining 22 feature sets, including the full feature set, the baseline feature set and a couple of manually selected feature sets. Unfortunately, due to size restrictions, not all features can be listed; though, indicative feature sets are listed in Table 5.

The most important results of the **classification approach** can be seen in Table 2 and the results of the **regression approach** in Tables 3 (development set) and 4 (shared task test set).

4 Discussion

4.1 Machine Learning Conclusions

Discrete classifiers (section 2.4.1) do not yield encouraging accuracy, as acceptable levels of accuracies appear only with a discretisation interval of 1.00, which though cannot be accepted due to its high Root Mean Square Error (RMSE). On the development keep-out set, the discretised Tree classifier seemingly outperforms all other methods (including the regression learners), since it yields a RMSE of 0.84, given several different feature vectors. Unfortunately, when applied to the final unknown test data, these classifiers performed obviously bad, providing the same single value for all sentences. We could attribute this to overfitting vs. sparse data and consider how we can handle this better in further work.

Another remarkable observation was the incapability of the RMSE to objectively show the quality of the model, in situations where the predicted values are very close or equal to the average of all real values. A Support Vector Machine with RMSE = 0.86 ranked 3rd among the classifiers, although it “cheated” by producing only the average value: 3.25. This leads to the conclusion that the selection of the best algorithm is not just dictated by the lowest RMSE, but it should consider several other indications such as the standard deviation.

We therefore resort to the **regression learners** (section 2.4.2), whose scores are not worse, having a RMSE of 0.855. We have to notice that the four

		avg. 70-30%, 90-10% folds			
algorithm	f. set	RMSE	MAE	interval	
PLS	**#19**	**0.86**	0.69	**2.5**	**4.3**
Lasso	#19	0.86	0.68	2.7	4.4
Linear	#19	0.86	0.68	2.6	4.5
MARS	#19	0.86	0.68	2.6	4.7
PLS	#18	0.86	0.69	2.7	4.4
Linear	#18	0.86	0.69	2.8	4.4
Lasso	#18	0.86	0.69	2.8	4.4
MARS	**#16**	0.87	0.69	**2.4**	**4.6**
MARS	#18	0.86	0.69	2.4	4.5
MARS	#4	0.86	0.69	3.4	4.5
PLS	#16	0.87	0.70	2.1	4.8
PLS	#4	0.87	0.70	2.1	5.4
Linear	#4	0.88	0.70	2.4	4.8
Linear	#16	0.88	0.70	1.4	4.9
Lasso	#4	0.88	0.70	1.9	5.3
MARS	#2	0.90	0.72	3.0	4.5
Lasso	#16	0.90	0.71	2.7	4.5
Linear	#2	0.90	0.72	3.0	4.0
Lasso	#2	0.90	0.72	3.0	4.0
PLS	#2	0.90	0.73	3.0	3.9
Tree	#21	1.08	0.86	1.5	5.0
Tree	#19	1.19	0.96	1.6	5.0
Tree	#16	1.23	0.98	1.6	5.0
Tree	#18	1.25	0.98	1.4	5.0

Table 3: Regression results. Root Mean Square Error (RMSE) and Mean Average Error (MAE), Largest Error Percentage (LEP) and Smallest Error Percentage (SEP). Bold face indicates submitted sets

regression algorithms have comparable performance given the same features.

The best-performing feature set (#19) which was chosen as the first submission (DFKI_cfs-plsreg) trained with PLS regression, contains features indicated by Correlation-based Feature Selection, run with *bestfirst* on a 10-fold cross-validation. We used the features which were selected on the 100% or 90% of the folds. An equally best-performing feature set (#18) has resulted from exactly the same feature selection execution, but contains only features which were selected in all folds.

The second submission (DFKI_grcfs-mars) was chosen to differentiate both the feature set and the learning method, with respect to a decent interval. Feature set #16 is the result of the Correlation-based Feature Selection, run in a *greedy-stepwise* mode. The regression was trained with MARS.

learner	feat.	name	RMSE	MAE
MARS	#16	grcfs–mars	0.98	0.82
PLS	#19	cfs-plsreg	0.99	0.82

Table 4: Results of the submitted methods on the official testset

The baseline feature set (#2) performed worse. Noticeable was the RMSE of the feature set #4, with features selected based on their *Gain Ratio*, but we did not submit this due to its very narrow interval.

4.2 Feature conclusions

The best performing feature set gives interesting hints on what worked as a best indication of translation quality. We would try to summarize them as follows:

- The language checking of the source sentence detected *complex* or *embedded sentences*, which are often not handled properly by SMT due to their complicated structure.

- The language checking of the target sentence detected several agreement issues.

- Parsing provided of source and target count of verbs, nouns, adjectives and secondary sentences; with the assumption that translations are relatively isomorphic, the loss of a verb or a noun or the inability to properly handle a secondary sentence, would mean a considerably bad translation outcome. The number of parse trees generated for each sentence can be an indication of ambiguity.

- Punctuation (dots, commas) often indicates a complex sentence structure.

- The most useful decoding features were the inverse phrase translation probability (a_1), the inverse lexical weighting (a_2), the phrase probability (pC) and future cost estimate (c) as well as statistics over their incremental values along the search graph.

feature			
set	type	source	target
#19	Baseline	LM, %bi_q$_4$, punct	LM, punct
	Checker	complex_sent, embedded_sent	pp_v_plural, nom_adj_masc
	Parsing	trees, CC, NP, NN, JJ, comma	trees, S, CC, VB, VP, NN, JJ, dot
	Decoding		avg(a_2), a_1, a_2
#16	Baseline	LM, seen, punct, %uni_q$_1$, %bi_q$_1$, %bi_q$_4$, %tri_q$_4$	LM, target_occ
	Checker	score: style, spelling, quality; verb: agr, form, obj_inf, close_to_subj; avoid_parenth, complex_sent, these_those_noun, np_num_agr, noun_adj_conf, repeat_subj, wrong_seq, wrong_word, disamb_that, use_rel_pron, use_article, avoid_dangling, repeat_modal, use_complement	double_punct, to_too_confusion, word_repeat, det_nom_sing, pp_v_plural, pp_v_sing, nom_adj_plural, comma_parenth_space, nom_adj_fem, nom_adj_masc, nom_adj_sing, det_nom_fem, del_nom_sing, del_nom_masc, det_nom_plur
	Parsing	trees, S, CC, JJ, comma, VB, NP, NN, VP	trees, S, CC, JJ, NP, VB, NN, VP, dot, PP
	Decoding		avg(pC), avg(a_1), std(pC), var(c), std(*lm*), avg(a_2), d$_2$, std(c), a_1, a_2

Table 5: Indicative feature sets for the most successful quality estimation models. Features explained at section 2.2

Acknowledgments

This work has been developed within the TaraXŰ project financed by TSB Technologiestiftung Berlin – Zukunftsfonds Berlin, co-financed by the European Union – European fund for regional development. Many thanks to Lukas Poustka for technical help on feature acquisition, to Melanie Siegel for the proprietary language checking tool, and to the reviewers for the useful comments.

References

Eleftherios Avramidis, Maja Popovic, David Vilar, Aljoscha Burchardt, and Maja Popović. 2011. Evaluate with Confidence Estimation : Machine ranking of translation outputs using grammatical features. *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 65–70, July.

Eleftherios Avramidis. 2011. DFKI System Combination with Sentence Ranking at ML4HMT-2011. In *Proceedings of the International Workshop on Using Linguistic Information for Hybrid Machine Translation (LIHMT 2011) and of the Shared Task on Applying Machine Learning Techniques to Optimising the Division of Labour in Hybrid Machine Translation (M. Sha*. Center for Language and Speech Technologies and Applications (TALP), Technical University of Catalonia.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. 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*, Montreal, Canada, June. Association for Computational Linguistics.

William S Cleveland. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836.

Janez Demšar, Blaz Zupan, Gregor Leban, and Tomaz Curk. 2004. Orange: From Experimental Machine Learning to Interactive Data Mining. In *Principles of Data Mining and Knowledge Discovery*, pages 537–539.

Jerome H. Friedman. 1991. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1–67, March.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level MT evaluation without reference translations : Beyond language modeling. *Language*, (2001):103–111.

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.

Mark A Hall. 2000. Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. In Pat Langley, editor, *Proceedings of 17th International Conference on Machine Learning*, pages 359–366. Morgan Kaufmann Publishers Inc.

Igor Kononenko. 1994. Estimating attributes: analysis and extensions of RELIEF. In *Proceedings of the European conference on machine learning on Machine Learning*, pages 171–182, Secaucus, NJ, USA. Springer-Verlag New York, Inc.

S Kullback and R A Leibler. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22:49–86.

M Antònia Martí Mariona Taulé and Marta Recasens. 2008. AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA).

Kristen Parton, Joel Tetreault, Nitin Madnani, and Martin Chodorow. 2011. E-rating Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 108–115, Edinburgh, Scotland, July. Association for Computational Linguistics.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *In HLT-NAACL 07*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.

Christopher B Quirk. 2001. Training a Sentence-Level Machine Translation Confidence Measure. *Evaluation*, pages 825–828.

Sylvain Raybaud, Caroline Lavecchia, David Langlois, and Kamel Smaïli. 2009. New Confidence Measures for Statistical Machine Translation. *Proceedings of the International Conference on Agents*, pages 394–401.

Felipe Sánchez-Martinez. 2011. Choosing the best machine translation system to translate a sentence by using only source-language information. In Mikel L Forcada, Heidi Depraetere, and Vincent Vandeghinste, editors, *Proceedings of the 15th Annual Conference of the European Associtation for Machine Translation*, number May, pages 97–104, Leuve, Belgium. European Association for Machine Translation.

Lucia Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. In *13th Annual Meeting of the European Association for Machine Translation (EAMT-2009)*, pages pp. 28–35, Barcelona, Spain.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904. ISCA, September.

M Stone and R J Brooks. 1990. Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. *Journal of the Royal Statistical Society Series B Methodological*, 52(2):237–269.

R Tibshirani. 1994. Regression shrinkage and selection via the lasso.

Nicola Ueffing and Hermann Ney. 2007. Word-Level Confidence Estimation for Machine Translation. *Computational Linguistics*, 33(1):9–40.