Quality Estimation: an experimental study using unsupervised similarity measures

Erwan Moreau
CNGL and Computational Linguistics Group
Centre for Computing and Language Studies
School of Computer Science and Statistics
Trinity College Dublin
Dublin 2, Ireland
moreaue@cs.tcd.ie

Carl Vogel
Computational Linguistics Group
Centre for Computing and Language Studies
School of Computer Science and Statistics
Trinity College Dublin
Dublin 2, Ireland
vogel@cs.tcd.ie

Abstract

We present the approach we took for our participation to the WMT12 Quality Estimation Shared Task: our main goal is to achieve reasonably good results without appeal to supervised learning. We have used various similarity measures and also an external resource (Google N-grams). Details of results clarify the interest of such an approach.

1 Introduction

Quality Estimation (or Confidence Estimation) refers here to the task of evaluating the quality of the output produced by a Machine Translation (MT) system. More precisely it consists in evaluating the quality of every individual sentence, in order (for instance) to decide whether a given sentence can be published as it is, should be post-edited, or is so bad that it should be manually re-translated.

To our knowledge, most approaches so far (Specia et al., 2009; Soricut and Echihabi, 2010; He et al., 2010; Specia et al., 2011) use several features combined together using supervised learning in order to predict quality scores. These features belong to two categories: *black box features* which can be extracted given only the input sentence and its translated version, and *glass box features* which rely on various intermediate steps of the internal MT engine (thus require access to this internal data). For the features they studied, Specia et al. (2009) have shown that *black box features* are informative enough and *glass box features* do not significantly contribute to the accuracy of the predicted scores.

In this study, we use only *black box features*, and further, eschew supervised learning except in the broadest sense. Our method requires some reference data, all taken to be equally good exemplars of a positive reference category, against which the experimental sentences are compared automatically. This is the extent of broader-sense supervision. The method does not require a training set of items each annotated by human experts with quality scores (except for the purpose of evaluation of course).

Successful unsupervised learning averts risks of the alternative: supervised learning necessarily makes the predicting system dependent on the annotated training data, i.e. less generic, and requires a costly human evalution stage to obtain a reliable model. Of course, our approach is likely not to perform as well as supervised approaches: here the goal is to find a rather generic robust way to measure quality, not to achieve the best accuracy. Nevertheless, in the context of this Quality Evaluation Shared task (see (Callison-Burch et al., 2012) for a detailed description) we have also used supervised learning as a final stage, in order to submit results which can be compared to other methods (see §4).

We investigate the use of various similarity measures for evaluating the quality of machine translated sentences. These measures compare the sentence to be evaluated against a *reference* text, providing a similarity score result. The reference data is supposed to represent standard (well-formed) language, so that the score is expected to reflect how complex (source side) or how fluent (target side) the given sentence is.

After presenting the similarity measures in sec-

tion 2, we will show in section 3 how they perform individually on the ranking task; finally we will explain in section 4 how the results that we submitted were obtained using supervised learning.

2 Approach

Our method consists in trying to find the best measure(s) to estimate the quality of machine translated sentences, i.e. the ones which show the highest correlation with the human annotators scores. The measures we have tested work always as follows.

Given a sentence to evaluate (source or target), a score is computed by comparing the sentence against a reference dataset (usually a big set of sentences). This dataset is assumed to represent standard and/or well-formed language.[1] This score represents either the quality (similarity measure) or the faultiness (distance measure) of the sentence. It is not necessarily normalized, and in general cannot be interpreted straightforwardly (for example like the 1 to 5 scale used for this Shared Task, in which every value 1, 2, 3, 4, 5 has a precise meaning). In the context of the Shared task, this means that we focus on the "ranking" evaluation measures provided rather than the "scoring" measures. These scores are rather intended to compare sentences relatively to one another: for instance, they can be used to discard the N% lowest quality sentences from post-editing.

The main interest in such an approach is in avoiding dependence on costly-to-annotate training data—correspondingly costly to obtain and which risk over-tuning the predicting system to the articulated features of the training items. Our method still depends on the dataset used as reference, but this kind of dependency is much less constraining, because the reference dataset can be any text data. To obtain the best possible results, the reference data has to be representative enough of what the evaluated sentences *should* be (if they were of perfect quality), which implies that:

- a high coverage (common words or n-grams) is preferable; this also means that the size of this dataset is important;
- the quality (grammaticality, language register, etc.) must be very good: errors in the reference data will infect the predicted scores.

It is rather easy to use different reference datasets with our approach (as opposed to obtain new human scores and training a new model on this data), since nowadays numerous textual resources are available (at least for the most common languages).

2.1 Similarity measures

All the measures we have used compare (in different ways) the n-grams of the tested sentence against the reference data (represented as a big *bag of n-grams*). There is a variety of parameters for each measure; here are the parameters which are common to all:

Length of n-grams: from unigrams to 6-grams;

Punctuation: with or without punctuation marks;

Case sensitivity: binary;

Sentence boundaries: binary signal of whether special tokens should be added to mark the start and the end of sentences.[2] This permits:

- that there is the same number of n-grams containing a token w, for every w in the sentence;
- to match n-grams starting/ending a sentence only against n-grams which start/end a sentence.

Most configurations of parameters presented in this paper are empirical (i.e. only the parameter settings which performed better during our tests were retained). Below are the main measures explored.[3]

2.1.1 Okapi BM25 similarity (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) is a widely used similarity measure in Information Retrieval(IR). It has also been shown to perform significantly better than only term frequency in tasks like matching coreferent named entities (see e.g. Cohen et al. (2003)), which is

[1] We use this definition of "reference" in this article. Please notice that this differs from the sense "human translation of a source sentence", which is more common in the MT literature.

[2] With trigrams, "`Hello World !`" (1 trigram) becomes "`# # Hello World ! # #`" (5 trigrams).

[3] One of the measures is not addressed in this paper for IP reasons (this measure obtained good results but was not best).

technically not very different from comparing sentences. The general idea is to compare two documents[4] using their bags of n-grams representations, but weighting the frequency of every n-gram with the IDF weight, which represents "how meaningful" the n-gram is over all documents based on its inverse frequency (because the n-grams which are very common are not very meaningful in general).

There are several variants of TF-IDF comparison measures. The most recent "Okapi BM25" version was shown to perform better in general than the original (more basic) definition (Jones et al., 2000). Moreover, there are different ways to actually combine the vectors together (e.g. L1 or L2 distance). In these experiments we have only used the Cosine distance, with Okapi BM25 weights. The weights are computed as usual (using the number of sentences containing X for any n-gram X), but are based only on the reference data.

2.1.2 Multi-level matching

For a given length N, "simple matching" is defined as follows: for every N-gram in the sentence, the score is incremented if this N-gram appears at least once in the reference data. The score is then relativized to the sentence N-gram length.

"Multi-level matching" (MLM) is similar but with different lengths of n-grams. For (maximum) length N, the algorithm is as follows (for every n-gram): if the n-gram appears in the reference data the score is incremented; otherwise, for all n-grams of length $N-1$ in this n-gram, apply recursively the same method, but apply a penalty factor p ($p < 1$) to the result.[5] This is intended to overcome the binary behaviour of the "simple matching". This way short sentences can always be assigned a score, and more importantly the score is smoothed according to the similarity of shorter n-grams (which is the behaviour one wants to obtain intuitively).

[4]In this case every sentence is compared against the reference data; from an IR viewpoint, one can see the reference data as the request and each sentence as one of the possible documents.

[5]This method is equivalent to computing the "simple matching" for different lengths N of N-grams, and then combine the scores s_N in the following way: if $s_N < s_{N-1}$, then add $p \times (s_{N-1} - s_N)$ to the score, and so on. However this "external" combination of scores can not take into account some of the extensions (e.g. weights).

Two main variants have been tested. The first one consists in using skip-grams.[6] Different sizes and configurations were tested (combining skip-grams and standard sequential n-grams), but none gave better results than using only sequential n-grams. The second variant consists in assigning a more fine-grained value, based on different parameters, instead of always assigning 1 to the score when n-gram occurs in the reference data. An optimal solution is not obvious, so we tried different strategies, as follows.

Firstly, **using the global frequency of the ngram in the reference data**: intuitively, this could be interpreted as "the more an n-gram appears (in the reference data), the more likely it is well-formed". However there are obviously n-grams which appear a lot more than others (especially for short n-grams). This is why we also tried using the logarithm of the frequency, in order to smooth discrepancies.

Secondly, **using the inverse frequency**: this is the opposite idea, thinking that the common n-grams are easy to translate, whereas the rare n-grams are harder. Consequently, the critical parts of the sentence are the rare n-grams: assigning them more weight focuses on these. This works in both cases (if the n-gram is actually translated correctly or not), because the weight assigned to the n-gram is taken into account in the normalization factor.

Finally, **using the Inverse Document Frequency (IDF)**: this is a similar idea as the previous one, except that instead of considering the global frequency the number of sentences containing the n-gram is taken into account. In most cases (and in all cases for long n-grams), this is very similar to the previous option because the cases where an n-gram (at least with $n > 1$) appears several times in the same sentence are not common.

2.2 Resources used as reference data

The reference data against which the sentences are compared is crucial to the success of our approach. As the simplest option, we have used the Europarl data on which the MT model was trained (source/target side for source/target sentences). Separately we tested a very different kind of data, namely the Google Books N-grams (Michel et al.,

[6]The `true-false-true` skip-grams in "`There is no such thing`": `There no`, `is such` and `no thing`.

2011): it is no obstacle that the reference sentences themselves are unavailable, since our measures only need the set of n-grams and possibly their frequency (Google Books N-gram data contains both).

3 Individual measures only

In this section we study how our similarity measures and the baseline features (when used individually) perform on the ranking task. This evaluation can only be done by means of DeltaAvg and Spearman correlation, since the values assigned to sentences are not comparable to quality scores. We have tested numerous combinations of parameters, but show below only the best ones (for every case).

3.1 General observations

Method	Ref. data	DeltaAvg	Spearman
MLM,1-4	Google, eng	0.26	0.22
Baseline feature 1		0.29	0.29
Baseline feature 2		0.29	0.29
MLM,1-3,lf	Google, spa	0.32	0.28
Okapi,3,b	EP, spa	0.33	0.27
Baseline feature 8		0.33	0.32
Okapi,2,b	EP, eng	0.34	0.30
Baseline feature 12		0.34	0.32
Baseline feature 5		0.39	0.39
MLM,1-5,b	EP, spa	0.39	0.39
MLM,1-5,b	EP, eng	0.39	0.40
Baseline feature 4		0.40	0.40

Table 1: Best results by method and by resource on training data. *b* = sentence boundaries ; *lf* = log frequency (Google) ; EP = Europarl.

Table 1 shows the best results that every method achieved on the whole training data with different resources, as well as the results of the best baseline features.[7] Firstly, one can observe that the language model probability (baseline features 4 and 5) performs as good or slightly better than our best measure. Then the best measure is the one which combines different lengths of n-grams (multi-level matching, combining unigrams to 5-grams), followed by baseline feature 12 (percentage of bigrams in quartile 4 of frequency), and then Okapi BM25 applied to bigrams. It is worth noticing that comparing either the source sentence or the target sentence (against the source/target training data) gives very similar results. However, using Google Ngrams as reference data shows a significantly lower correlation. Also using skip-grams or any of our "fined-grained" scoring techniques (see §2.1.2) did not improve the correlation, even if in most cases these were as good as the standard version.

3.2 Detailed analysis: how measures differ

Even when methods yield strongly correlated results, differences can be significant. For example, the correlation between the rankings obtained with the two best methods (baseline 4 and MLM Eng.) is 0.53. The methods do not make the same errors.[8] A method may tend to make a lot of small errors, or on the contrary, very few but big errors.

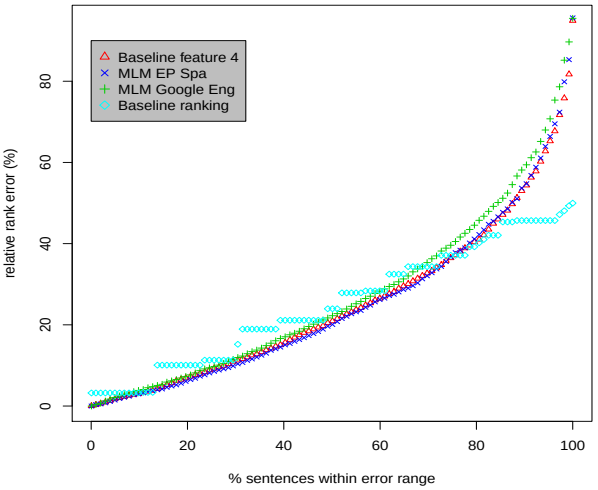

Figure 1: Percentage of best segments within an error range. For every measure, the X axis represents the sentences sorted by the difference between the predicted rank and the actual rank ("rank error"), in such a way that for any (relative) number of sentences x, the y value represents the maximum (relative) rank error for all prior sentences: for instance, 80% of the ranks predicted by these three measures are at most 40% from the actual rank.

Let R and R' be the actual and predicted ranks[9] of sentence, respectively. Compute the difference

[7] Baseline 1,2: length of the source/target sentence;
Baseline features 4,5: LM probability of source/target sentence;
Baseline feature 8: average number of translations per source word with threshold 0.01, weighted by inverse frequency;
Baseline feature 12: percentage of bigrams in quartile 4 of frequency of source words in a corpus of the source language.

[8] This motivates use of supervised learning (but see §1).

[9] It is worth noticing that ties are taken into account here: two

$D = |R - R'|$; then relativize to the total number of sentences (the upper bound for D): $D' = D/N$. D' is the *relative rank error*. On ascending sort by D', the predicted ranks for the first sentences are closest to their actual rank. Taking the relative rank error D'_j for the sentence at position M_j, one knows that all "lower" sentences ($\forall M_i$, $M_i \leq M_j$) are more accurately assigned ($D'_i \leq D'_j$). Thus, if the position is also relativized to the total number sentences: $M'_k = M_k/N$, M'_k is the proportion of sentences for which the predicted rank is at worst $D'_k\%$ from the real rank. Figure 1 shows the percentage of sentences withing a rank error range for three good methods:[10] the error distributions are surprisingly similar. A *baseline ranking* is also represented, which shows the same if all sentences are assigned the same rank (i.e. all sentences are considered of equal quality)[11].

We have also studied effects of some parameters:

- Taking punctuation into account helps a little;
- Ignoring case gives slightly better results;
- Sentences boundaries significantly improve the performance;
- Most of the refinements of the local score (frequency, IDF, etc.) do not perform better than the basic binary approach.

4 Individual measures as features

In this section we explain how we obtained the submitted results using supervised learning.

4.1 Approach

We have tested a wide range of regression algorithms in order to predict the scores, using the Weka[12] toolkit (Hall et al., 2009). All tests were done using the whole training data in a 10 folds cross-validation setting. The main methods were:

- Linear regression
- Pace regression (Wang and Witten, 2002)
- SVM for regression (Shevade et al., 2000) (`SMOreg` in Weka)
- Decision Trees for regression (Quinlan, 1992) (`M5P` in Weka)

We have tested several combinations of features among the features provided as baseline and our measures. The measures were primarily selected on their individual performance (worst measures were discarded). However we also had to take the time constraint into account, because some measures require a fair amount of computing power and/or memory and some were not finished early enough. Finally we have also tested several attributes selection methods before applying the learning method, but they did not achieve a better performance.

4.2 Results

Table 2 shows the best results among the configurations we have tested (expressed using the official evaluation measures, see (Callison-Burch et al., 2012) for details). These results were obtained using the default Weka parameters.In this table, the different features sets are abbreviated as follows:

- **B**: Baseline (17 features);
- **M1**: All measures scores (45 features);
- **M2**: Only scores obtained using the provided resources (33 features);
- **L**: Lengths (of source and target sentence, 2 features).

For every method, the best results were obtained using all possible features (baseline and our measures). The following results can also be observed:

- our measures increase the performance over use of baseline features only (B+M1 vs. B);
- using an external resource (here Google n-grams) with some of our measures increases the performance (B+M1 vs. B+M2);

sentences which are assigned the same score are given the same rank. The ranking sum is preserved by assigning the average rank; for instance if $s_1 > s_2 = s_3 > s_4$ the corresponding ranks are 1, 2.5, 2.5, 4).

[10] Some are not shown, because the curves were too close.

[11] Remark: the plateaus are due to the ties in the *actual* ranks: there is one plateau for each score level. This is not visible on the predicted rankings because it is less likely that an important number of sentences have both the same actual rank and the same predicted rank (whereas they all have the same "predicted" rank in the baseline ranking, by definition).

[12] `www.cs.waikato.ac.nz/ml/weka` – l.v., 04/2012.

Features	Method	DeltaAvg	Spearman	MAE	RMSE
B	SVM	0.398	0.445	0.616	0.761
B	Pace Reg.	0.399	0.458	0.615	0.757
L + M1	SVM	0.401	0.439	0.615	0.764
L + M1	Lin. Reg	0.408	0.441	0.610	0.757
B	Lin. Reg.	0.408	0.461	0.614	0.754
L + M1	M5P	0.409	0.441	0.610	0.757
B + M2	SVM	0.409	0.447	0.605	0.753
B + M2	Pace Reg.	0.417	0.466	0.603	0.744
B + M2	M5P	0.419	0.472	0.601	0.746
L + M1	Pace Reg.	0.426	0.454	0.603	0.751
B + M2	Lin. Reg.	0.428	0.481	0.598	0.740
B	M5P	0.434	0.487	0.586	0.729
B + M1	SVM	0.444	0.489	0.585	0.734
B + M1	Pace Reg.	0.453	0.505	0.584	0.724
B + M1	Lin. Reg.	0.456	0.507	0.583	0.724
B + M1	M5P	0.457	0.508	0.583	0.724

Table 2: Best results on 10-folds cross-validation on the training data (sorted by DeltaAvg score).

- the baseline features contribute positively to the performance (B+M1 vs. L+M1);
- The M5P (Decision trees) method works best in almost all cases (3 out of 4).

Based on these training results, the two systems that we used to submit the test data scores were:

- **TCD-M5P-resources-only**, where scores were predicted from a model trained using M5P on the whole training data, taking only the baseline features (B) into account;
- **TCD-M5P-all**, where scores were predicted from a model trained using M5P on the whole training data, using all features (B+M1).

The **TCD-M5P-resources-only** submission ranked 5th (among 17) in the ranking task, and 5th among 19 (tied with two other systems) in the scoring task (Callison-Burch et al., 2012). Unfortunately the **TCD-M5P-all** submission contained an error.[13] Below are the official results for **TCD-M5P-resources-only** and the corrected results for **TCD-M5P-all** :

[13]In four cases in which Google n-grams formed the reference data, the scores were computed using the wrong language (Spanish instead of English) as the reference. Since this error occured only for the test data (not the training data used to compute the model), it made the predictions totally meaningless.

Submission	DeltaAvg	Spearman	MAE	RMSE
resources-only	0.56	0.58	0.68	0.82
all	0.54	0.54	0.70	0.84

Contrary to previous observations using the training data, these results show a better performance without our measures. We think that this is mainly due to the high variability of the results depending on the data, and that the first experiments are more significant because cross-validation was used.

5 Conclusion

In conclusion, we have shown that the robust approach that we have presented can achieve good results: the best DeltaAvg score reaches 0.40 on the training data, when the best supervised approach is at 0.45. We think that this robust approach complements the more fine-grained approach with supervised learning: the former is useful in the cases where the cost to use the latter is prohibitive.

Additionally, it is interesting to see that using external data (here the Google N-grams) improves the performance (when using supervised learning). As future work, we plan to investigate this question more precisely: when does the external data help? What are the differences between using the training data (used to produce the MT engine) and another dataset? How to select such an external data in order to maximize the performance? In our unsupervised framework, is it possible to combine the score obtained with the external data with the score obtained from the training data? Similarly, can we combine scores obtained by comparing the source side and the target side?

Acknowledgments

This research is supported by Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) funding at Trinity College, University of Dublin.
We thank the organizers who have accepted to apply a bug-fix (wrong numbering of the sentences) in the official results, and for organizing the Shared task.

References

[Callison-Burch et al.2012] 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.

[Cohen et al.2003] W.W. Cohen, P. Ravikumar, and S.E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, pages 73–78.

[Hall et al.2009] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. 2009. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

[He et al.2010] Y. He, Y. Ma, J. van Genabith, and A. Way. 2010. Bridging smt and tm with translation recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 622–630. Association for Computational Linguistics.

[Jones et al.2000] Karen Sparck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments - parts 1 and 2. *Inf. Process. Manage.*, 36(6):779–840.

[Michel et al.2011] J.B. Michel, Y.K. Shen, A.P. Aiden, A. Veres, M.K. Gray, J.P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176.

[Quinlan1992] J.R. Quinlan. 1992. Learning with continuous classes. In *Proceedings of the 5th Australian joint Conference on Artificial Intelligence*, pages 343–348. Singapore.

[Shevade et al.2000] S.K. Shevade, SS Keerthi, C. Bhattacharyya, and K.R.K. Murthy. 2000. Improvements to the smo algorithm for svm regression. *Neural Networks, IEEE Transactions on*, 11(5):1188–1193.

[Soricut and Echihabi2010] R. Soricut and A. 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. Association for Computational Linguistics.

[Specia et al.2009] Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Conference of the European Association for Machine Translation*, pages 28–35.

[Specia et al.2011] L. Specia, N. Hajlaoui, C. Hallett, and W. Aziz. 2011. Predicting machine translation adequacy. In *Machine Translation Summit XIII*, Xiamen, China.

[Wang and Witten2002] Y. Wang and I.H. Witten. 2002. Modeling for optimal probability prediction. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 650–657. Morgan Kaufmann Publishers Inc.