MTWatch: A Tool for the Analysis of Noisy Parallel Data

Sandipan Dandapat[1], **Declan Groves**[2]

[1]Department of Computer Science and Engineering
IIT Guwahati, Assam, India
[2]Centre for Next Generation Localisation, School of Computing
Dublin City University, Dublin, Ireland
sdandapat@iitg.ernet.in, dgroves@computing.dcu.ie

Abstract

State-of-the-art statistical machine translation (SMT) technique requires a good quality parallel data to build a translation model. The availability of large parallel corpora has rapidly increased over the past decade. However, often these newly developed parallel data contains contain significant noise. In this paper, we describe our approach for classifying good quality parallel sentence pairs from noisy parallel data. We use 10 different features within a Support Vector Machine (SVM)-based model for our classification task. We report a reasonably good classification accuracy and its positive effect on overall MT accuracy.

Keywords: Statistical Machine Translation, Noisy Parallel Data, Quality Estimation

1. Introduction

The success of a state-of-the-art SMT system is highly dependent on the amount of suitable parallel corpora available for training the engine. Large amounts of good quality parallel corpora (OPUS (Tiedemann and Nygaard., 2004), Europarl (Koehn., 2005), etc.) are available for English and other European Languages. However, the amounts of available domain-specific parallel data is still in its infancy for many dominant language pairs and for several domains (Axelrod et al., 2011). In addition, there exist a large number of languages which suffer from the scarcity of reasonably good amounts of parallel corpora e.g. Indic Languages, etc. (Dandapat et al., 2011) .

Due to the rapid growth of multilingual web documents, parallel corpora are more frequently created from these bilingual resources for many language pairs, particularly for those languages (and domains) that are under-resourced. However, it is difficult to ensure or measure the quality of these resources and their suitability for training MT systems. They may constitute comparable, rather than truly parallel corpora, and often multilingual versions of these web-based documents have themselves been generated using MT without any human post-editing.

More recently, as an alternative to web-crawling, crowd sourcing has been effectively used to create parallel data (Zaidan and Callison-Burch., 2011; Post et al., 2012), especially for a particular domain within a short period of time (Lewis et al., 2011). However, crowd-based techniques also carry the risk of returning noisy data, due to the difficulty in controlling the crowd, in terms of producing high-quality translations.

In this work, we account for the various types of noise that may be introduced to parallel corpora and report a method for automatically classifying sentence pairs into 'good' and 'bad' instances.

Very little work has been done previously on automatic quality estimation of parallel data. Huerta (2011) describes their study on different metrics for automatic quality estimation and validation of manually-developed parallel corpora. Along with some of their features, we have used additional features in a classifier-based model for the quality estimation task. Feature-based classification has been widely used for MT confidence estimation (Specia et al., 2009; He et al., 2010); our work uses a similar concept of using features for automatic classification of manually developed parallel corpora.

Goutte et al. (2012) studied the effect of noisy data in MT quality. They report that performance in MT only begins to degrade when 30% noise or more is introduced. They used very large data sets which are often not available for various domains. Filtering noisy sentences is a more significant issue when we have less data at our disposal, as is the case when building domain-specific systems.

2. Our Approach

We use an SVM-based model to classify the noisy data into two classes; *good* and *bad* data. We trained an SVM model based on noisy parallel data using a potential feature set. Consider $D_n(D_n^s, D_n^t)$ is the noisy parallel data. (d_i^s, d_i^t) is a source-target translation pair where $d_i^s \in D_n^s$ and $d_i^t \in D_n^t$. We estimate the value for each of the features for all (d_i^s, d_i^t) in D_n. The feature vector corresponding to a particular sentence pair and the associated label (*good* or *bad*) is used as an instance for the classification task.

2.1. Features Used

We use a total of ten different features for the classifier. The majority of the features we have used are completely language-independent. First, we derive three features explicit to the source-target sentence pair (d_i^s, d_i^t) in the noisy parallel data:

Length Ratio: This feature estimates the normalized length ratio between the source- and the target sentence. The idea is that the length of the translation of a source sentence should not diverge too much if they are coarsely

equivalent.
$f_1(d_i^s, d_i^t) = \frac{|length(d_i^s) - length(d_i^t)|}{\max(|d_i^s|, |d_i^t|)}$

Marker Word Ratio: *Marker words* (Green, 1979) are closed category words or morphemes used to define the syntax of a language. Thus the marker word ratio is a good indicator of the distribution of functional word/morphemes between two languages. The traditional functional word is essentially a subset of the marker word set (which additionally includes morphemes and punctuation) of a language. The use of marker-word techniques is language dependant, however marker word lists are readily available for multiple languages and can be easily compiled for new languages, including those that are morphologically rich.
$f_2(d_i^s, d_i^t) = \frac{|MW(d_i^s) - MW(d_i^t)|}{\max(|MW(d_i^s)|, |MW(d_i^t)|)}$ where, $MW(s)$ denotes the number of marker words presents in s.

Marker Chunk Ratio: Marker chunks (Gough and Way., 2004) starts with a marker word and must contain at least one nonmarker (content) word. The idea behind using marker chunk ratio is that they often capture shallow syntactic units of a sentence. A marker chunk may contain more than one marker words thus even when the marker word distribution varies significantly, it is anticipated that the number of marker chunks should not differ largely.
$f_3(d_i^s, d_i^t) = \frac{|MC(d_i^s) - MC(d_i^t)|}{\max(|MC(d_i^s)|, |MC(d_i^t)|)}$ where, $MC(s)$ denotes the number of marker chunks presents in s.

It is often the case that there exists an MT system for a language pair build on a specific data. However, we may wish to add new parallel data to the system to improve both accuracy and coverage. Thus our particular attempt will help users to predict the goodness of the new parallel data before retraining the MT system blindly adding the new parallel data. The feature described bellow uses the source sentence d_i^s and its machine translated output $\bar{d}_i^t$. We used in house English–French MT system[1] to produce $\bar{d}_i^t$ for each source sentence d_i^s. Thus our second set of features are estimated based on the (d_i^s, d_i^t) and $\bar{d}_i^t$.

Length Ratio$_{\mathbf{MT}}$: This estimates the length ratio between d_i^s and $\bar{d}_i^t$.
$f_4(d_i^s, \bar{d}_i^t) = \frac{|length(d_i^s) - length(\bar{d}_i^t)|}{\max(|d_i^s|, |\bar{d}_i^t|)}$

Marker Word Ratio$_{\mathbf{MT}}$: Estimates the normalized ratio of marker words between d_i^s and $\bar{d}_i^t$.
$f_5(d_i^s, \bar{d}_i^t) = \frac{|MW(d_i^s) - MW(\bar{d}_i^t)|}{\max(|MW(d_i^s)|, |MW(\bar{d}_i^t)|)}$

Edit Distance: We use the edit distance (Wagner and Fischer., 1974) between d_i^t and $\bar{d}_i^t$ as another feature. It is likely that the machine translation $\bar{d}_i^t$ and the actual target translation d_i^t should have some overlap in surface words. It is anticipated that the overlap will be higher when d_i^t is noise free compared to a bad noisy translation. Thus, we use a normalized edit-distance score as a feature to the classifier.
$f_6(d_i^t, \bar{d}_i^t) = \frac{\mathrm{ED}(\mathrm{d}_\mathrm{i}^\mathrm{t}, \bar{\mathrm{d}}_\mathrm{i}^\mathrm{t})}{\max(|d_i^t|, |\bar{d}_i^t|)}$
where $\mathrm{ED}(x, y)$ refers to the word-based edit distance between x and y.

TER Score: We measure the translation edit rate (TER)[2] score between d_i^t and $\bar{d}_i^t$. Higher TER score indicates larger dissimilarity between the strings and vice versa.

Furthermore, we use three binary valued features for capturing capturing the noise text. These features includes:

URL: Estimates the presence and absence of URL in the candidate pair (d_i^s, d_i^t).

UTF8: Estimates the presence and absence of characters outside the UTF8 range of the particular language in d_i^t.

Punctuation Count (PC): Estimates whether an equal number of punctuation markers are present in the source- and target-side candidate sentence pair(d_i^s, d_i^t).

Thus we have a total of 10 features used to train a SVM model. Table 1 lists all features used for the classifier. These features are estimated from the noisy labelled data. We used labelled data to learn this supervised SVM model which is subsequently used to classify the new unseen data.

Features derived from (d_i^s, d_i^t)		
f_1	Length Ratio	$\frac{|length(d_i^s) - length(d_i^t)|}{\max(|d_i^s|, |d_i^t|)}$
f_2	Marker Words Ratio	$\frac{|MW(d_i^s) - MW(d_i^t)|}{\max(|MW(d_i^s)|, |MW(d_i^t)|)}$
f_3	Marker Chunk Ratio	$\frac{|MC(d_i^s) - MC(d_i^t)|}{\max(|MC(d_i^s)|, |MC(d_i^t)|)}$
Features derived from (d_i^s, d_i^t)		
f_4	Length Ratio	$\frac{|length(d_i^s) - length(\bar{d}_i^t)|}{\max(|d_i^s|, |\bar{d}_i^t|)}$
f_5	Marker Words Ratio	$\frac{|MW(d_i^s) - MW(\bar{d}_i^t)|}{\max(|MW(d_i^s)|, |MW(\bar{d}_i^t)|)}$
f_6	Edit Distance (ED)	$\frac{\mathrm{ED}(\mathrm{d}_\mathrm{i}^\mathrm{t}, \bar{\mathrm{d}}_\mathrm{i}^\mathrm{t})}{\max(|d_i^t|, |\bar{d}_i^t|)}$
f_7	TER Score (Snover et al., 2006)	translation edit rates between d_i^t and $\bar{d}_i^t$
Binary values features from (d_i^s, d_i^t)		
f_8	URL	same URL in d_i^s and d_i^t
f_9	UTF8	there exist character outside UTF8 range
f_{10}	Punctuation Count	equal number of punctuations present in d_i^s and d_i^t

Table 1: Features used for the classifier. $MW(s)$ and $MC(s)$ denotes the number of marker words and marker chunks presents in s respectively.

3. Parallel Corpora and Noise

In order to build the classifier, we need both positive and negative instances of parallel sentence pairs. As we did not have access to pre-classified data, we automatically created noisy data based on studying the different errors that tend to occur within parallel data, including those introduced by

[1] The English-French MT system was build using OpenMaTrEx (Dandapat et al., 2010) using 100k parallel sentences from Europarl (Koehn., 2005). We consider this as the baseline system to evaluate the classified data by putting these classified data into this existing 100k data.

[2] http://www.cs.umd.edu/ snover/tercom/

translators. While creating our noisy parallel data, we ensure that 50% of our data consisted of good quality parallel sentence pairs. We then introduce noise to the remaining 50%. This is to ensure that both positive and negative instances of the parallel sentence pairs are evenly distributed so that the classifier is not biased towards one class. The three different noise types that we introduced are described below:

Random Translation (N_1): This is to model the scenario where a translator may insert some random text in the target language as a translation of the source sentence. This situation often occurs in crowd-based environments when untrusted users insert a random target language sentence copied from other sources. This situation may also occur when extracting parallel data from comparable corpora through web crawling. In both scenarios, the target-side sentence is a grammatically correct sentence in the target language, but is not a translation equivalent of the respective source sentence. We randomly select English source-sentences (s_i) from Europarl data and randomly select some target sentences (t_i) from the target side of the Europarl such that $i \neq j$. Note that the source side sentences used for one particular purpose are disjoint from all other data.

Partial Translation (N_2): For generating this type of noise, for a given parallel sentence pair (s_i, t_i), we randomly remove some words from t_i to make it a noisy translation for the source sentence s_i. This particular noise essentially captures ill-formed sentences of the target language which partially translate the source sentence s_i. For our noise embedding process, we randomly remove 40% of words from the original t_i. Thus the length of the new noisy target translation is $len(t_i) - \lfloor 0.4len(t_i) \rfloor$.

Translation Using MT System (N_3): It is often the case that MT systems are used to produce parallel data with some post-editing effort. However, in the process of crowd-based parallel data collection, untrusted users often solely use the output of an MT system as the target translation, which can frequently produce lower quality translations. This noise is very hard to detect due to the increasing success of machine translation.[3] This also reflects the scenario where increasing quantities of multilingual data published on the web is generated using raw MT. For this process, we use Microsoft's Bing English–French MT system to produce the MT output for a randomly selected s_i from the source-side of English–French Europarl corpus.

4. Experiment and Results

In our experiments we tested both the accuracy of our SVM-classifier and evaluated the performance of our MT system when trained on different data sets, as selected by the classifier, in order to measure the effect of noise on translation quality. For all of our experiments we made use of data taken from the English–French section of the Europarl corpus (Koehn., 2005).

4.1. Classification Accuracy

The accuracy of the classifier is defined by the ratio of the instances correctly classified against the total number of instances. For this approach we took two different combinations of the noise types in order to estimate their effect on the classification task.

- We consider all three noise types along with the good data. All noisy instances are merged into a single error class. Thus, the classification task is evaluated as a binary classifier. We shall refer to this as SVMALL.
- Furthermore, we excluded the noise generated using the MT system (N_3) and only considered the remaining two classes of noise (N_1 and N_2) along with the good data. We shall refer to this as SVMALL−MT.

In order to train and test the classifier, we took 15k sentences pairs from the Europarl corpus as positive instances and randomly selected 5k sentences pairs to generate each noisy set (N_1, N_2 and N_3). These amounts ensure equal distribution of positive (15k) and negative (15k) instances of parallel data in the experimental data set. We used a 5-fold cross validation strategy to report the classification accuracy.

4.1.1. Results

Table 2 summarises the accuracy of our classifier for SVMALL and SVMALL−MT experiments. We find that the overall accuracy of the classifier is 80.17% when considering all types of noise. This is due to the lower accuracy in identifying the noisy class N_3, as machine translated text is difficult to distinguish from human translation. However, in the third column of the Table 2, we see that the classifier has quite a high overall accuracy (92.65%) when ignoring the MT based noise N_3. Altogether, the classifier has reasonably good accuracy for both N_1 and N_2 classes.

	SVMALL	**SVMALL−MT**
#Data Used	30k	20k
All	80.17	92.65 %
Positive	85.09	90.85 %
N_1	94.08	96.26%
N_2	95.04	92.64%
N_3	36.62	–

Table 2: Classifier accuracies with 5-fold cross validation for different classes.

4.2. MT Quality

For these experiments we wished to measure the effect of adding different data sets (both clean and noisy) on the translation quality of our MT system.

We first built a baseline English–French MT system using OpenMaTrEx (Dandapat et al., 2010), built on 100k parallel English–French sentence pairs.[4] We then took an additional (disjoint) portion of the Europarl data, consisting of

[3] We consider machine translated output as bad instances of parallel data, as we wish to focus on selecting human-quality translations for training our engines. However, MT technology often produces good quality translation for certain sentences.

[4] Note that we used only the target side of the parallel data to build a separate language model for each of our experiments.

120k parallel sentences. From this additional data set we randomly selected 60k sentence pairs and embedded our three different noise types, with even distribution (i.e. 20k instances for each noise class). This produced a data set consisting of 60k positive instances and 60K negative instances.

We then retrained the MT models by adding the following different data sets to the initial baseline training set:

- **Mixed Noisy Data (MD):** Both good and bad instances (the full 120k sentence pairs)
- **Clean Data (CD):** Actual positive instances from the MD data set (instances which we know are to be of good quality) (60k sentence pairs)
- **Classified Data (CL):** Pairs of sentences which are classified as good data according to our SVM classifier
- **Size Constrained Mixed Data** ($\mathrm{MD}_{\mathrm{size}}$)**:** A randomly selected subset of MD consisting of the same number of sentence pairs as contained in CL

We conduct experiments using the four different data sets described above, with three different combinations of noise types (using $N_1 + N_2 + N_3$, $N_1 + N_2$ and only N_1). We measure the BLEU (Papineni et al., 2002) score for all our experiments to see the effect of the classified data in translation performance. A completely disjoint set of 2k sentences pairs has been used to evaluate the MT system performance at each stage.

4.2.1. Results

Table 3 shows the impact of the classified data on translation accuracy under the different experimental setups. We can see that using the CL data set produces slight gains in translation performance, when compared to the addition of $\mathrm{MD}_{\mathrm{size}}$. However, making use of the entire MD data set results in higher translation quality, in terms of BLEU, compaired to all other scenarios. This is possibly attributed to the greater amount of data contained in MD which provides the system with additional useful phrase pairs present in both the clean and noisy sections of the parallel corpus (similar to the observation reported in (Goutte et al., 2012)).

# Training Data	Data Type	BLEU (%)
Initial MT System		
100k	Europarl Data	31.49
Noise N_1+N_2+N_3:: Classifier Accuracy 80.53%		
+120k	MD	33.17
+60k	CD	32.42
+66k	CL	32.58
+66k	$\mathrm{MD}_{\mathrm{size}}$	32.34
Noise N_1+N_2 :: Classifier Accuracy 93.29%		
+80k	MD	32.37
+40k	CD	31.99
+39k	CL	32.14
+39k	$\mathrm{MD}_{\mathrm{size}}$	31.87
Noise N_1 :: Classifier Accuracy 96.81%		
+40k	MD	32.20
+20k	CD	31.94
+19k	CL	31.93
+19k	$\mathrm{MD}_{\mathrm{size}}$	31.7

Table 3: Translation accuracy after adding noisy data and classified good quality data. '+' indicates the amount of data added into the initial data.

5. Observations

We found that the classifier is able to achieve a high degree of accuracy in classifying the noise introduced to the parallel data, except for the noisy class N_3. However, the data classified by our model as good has always shown improvement in terms of translation quality (minimum of 0.23 BLEU points), compared to the actual clean data when the amounts of additional sentences pairs remain same. This shows that it is more favourable to use the classifier for training data selection than to rely on only using those sentence pairs pre-classified as being clean.

The highest MT system performance is observed when the entire data set (MD) is used (we see a maximum improvement of 0.59 BLEU points compared to the accuracy using CL data set). However, it must be noted that making use of the larger training data set results in increasing the training time overhead of the system. This issue is compounded when moving towards using much larger data sets.

MT quality also depends on the amount of noise present in the data. It may be the case that for a reasonably noisy data the effect will not be significant in terms of the gains achieved in translation quality. We made use of a relatively small data set (100k sentences pairs) to build our initial system. Thus, the MT system can extract good quality phrase pairs from noisy data based on the initial distribution. The effect changes when a small amount of data is used for the initial system, which is often the case when building and collecting parallel data for a new language pair or for a narrow domain. This is shown in Table 4. From this table we see that when equal amounts of data are used (and no MT-based noise is introduced), the addition of the CL data set has better MT performance than compared to the use of additional MD data set.

# Training Data	Data Type	BLEU (%)
Initial MT System		
40k	Europarl Data	29.86
No MT-based noise		
+60k	Actual Clean Data (CD)	30.35
+66k	Classified as good (CL)	30.58
+66k	Mixed Data (CD+N1+N2)	30.30

Table 4: Translation accuracy after adding noisy data and classified good quality data.

6. Conclusion and Outlook

In this paper, we have shown how different features can be used to build a classifier for extracting good quality parallel data for MT from potentially noisy parallel data. We have shown that the classifier works with high accuracy (except for filtering out MT output) and have shown subsequently that using the classified data can result in gains in MT performance. For our work to date, we have made use of synthetically-generated noisy data due to the lack of availability of real noisy parallel data.

For our future work, we plan to use real noisy parallel data collected via crowd-sourcing methods to more fully understand and evaluate the effect of the proposed approach. Due to the accuracy of the classifier, we believe that this method will be particularly useful for filtering out bad quality crowd-sourced translations and in helping to identify and build a trusted crowd; something which is vital in using crowd-source translations in production environments.

7. Acknowledgements

This work is supported by Science Foundation Ireland (Grants SFI11-TIDA-B2040 and 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University.

8. References

A. Axelrod, X. He, and J. Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical in Natural Language Processing (EMNLP 2011)*, pages 355–362, Edinburgh, Scotland.

S. Dandapat, M. Forcada, D. Groves, S. Penkale, J. Tinsley, and A. Way. 2010. Openmatrex: A free/open-source marker-driven example-based machine translation system. In *Proceedings of the 7th International Conference on Natural Language Processing (IceTAL 2010)*, pages 121–126, Reykjavík, Iceland.

S. Dandapat, S. Morriessy, A. Way, and M. Forcada. 2011. Using example-based mt to support statistical mt when translating homogeneous data in resource-poor settings. In *Proceedings of the 15th Annual Meeting of the European Association for Machine Translation (EAMT 2011)*, pages 201–208, Leuven, Belgium.

N. Gough and A. Way. 2004. Robust large-scale ebmt with marker-based segmentation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2004)*, pages 95–104, Baltimore, MD.

C. Goutte, M. Carpuat, and G. Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *Proceedings of the 10th Biennial Conference of the Association for Machine Translation in the Americas (AMTA 2012)*, San Diego, CA.

T. Green. 1979. Robust large-scale ebmt with marker-based segmentation. *Journal of Verbal Learning and Behaviour*, 18:481–496.

Y. He, Y. Ma, J. Roturier, A. Way, and J. van Genabith. 2010. Improving the post-editing experience using translation recommendation: A user study. In *Proceedings of the 9th Conference of the Association for Machine Translation in Americas (AMTA 2010)*, pages 247–256, Denver, CO.

J. M. Huerta. 2011. Approaches to automatic quality estimation of manual translation in crowdsourcing parallel corpora development: A quality equivalence and cohort-consensus approach. Ibm research division, t. j. watson research center, ny.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X: The 10th Machine Translation Summit*, pages 79–86, Phuket, Thailand.

W. Lewis, R. Munro, and S. Vogel. 2011. Crisis mt: Developing a cookbook for mt in crisis situation. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 501–511, Edinburgh, Scotland.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 311–318, Philadelphia, PA.

M. Post, C. Callison-Burch, and M. Osborne. 2012. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 401–409, Montréal, Canada.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in Americas (AMTA 2006)*, pages 223–231, Cambridge, MA.

L. Specia, M. Turqui, Z. Wang, J. Shawe-Taylor, and C. Saunders. 2009. Improving the confidence of machine translation quality estimates. In *Proceedings of the 12th Machine Translation Summit (MT Summit XII)*, pages 136–143, Ottawa, Canada.

J. Tiedemann and L. Nygaard. 2004. The opus corpus - parallel and free. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1183–1186, Lisbon, Portugal.

R. Wagner and M. Fischer. 1974. The string to string correction problem. *Journal of Association of Computing Machinery (JACM)*, 21(1):168–173.

O. Zaidan and C. Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computation Linguistics (ACL 2011)*, pages 1220–1229, Portland, OR.