Goodness: A Method for Measuring Machine Translation Confidence

Nguyen Bach*
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
nbach@cs.cmu.edu

Fei Huang and Yaser Al-Onaizan
IBM T.J. Watson Research Center
1101 Kitchawan Rd
Yorktown Heights, NY 10567, USA
{huangfe, onaizan}@us.ibm.com

Abstract

State-of-the-art statistical machine translation (MT) systems have made significant progress towards producing user-acceptable translation output. However, there is still no efficient way for MT systems to inform users which words are likely translated correctly and how confident it is about the whole sentence. We propose a novel framework to predict word-level and sentence-level MT errors with a large number of novel features. Experimental results show that the MT error prediction accuracy is increased from 69.1 to 72.2 in F-score. The Pearson correlation between the proposed confidence measure and the human-targeted translation edit rate (HTER) is 0.6. Improvements between 0.4 and 0.9 TER reduction are obtained with the n-best list reranking task using the proposed confidence measure. Also, we present a visualization prototype of MT errors at the word and sentence levels with the objective to improve post-editor productivity.

1 Introduction

State-of-the-art Machine Translation (MT) systems are making progress to generate more usable translation outputs. In particular, statistical machine translation systems (Koehn et al., 2007; Bach et al., 2007; Shen et al., 2008) have advanced to a state that the translation quality for certain language pairs (e.g. Spanish-English, French-English, Iraqi-English) in certain domains (e.g. broadcasting news, force-protection, travel) is acceptable to users.

However, a remaining open question is how to predict confidence scores for machine translated words and sentences. An MT system typically returns the best translation candidate from its search space, but still has no reliable way to inform users which word is likely to be correctly translated and how confident it is about the whole sentence. Such information is vital to realize the utility of machine translation in many areas. For example, a post-editor would like to quickly identify which sentences might be incorrectly translated and in need of correction. Other areas, such as cross-lingual question-answering, information extraction and retrieval, can also benefit from the confidence scores of MT output. Finally, even MT systems can leverage such information to do n-best list reranking, discriminative phrase table and rule filtering, and constraint decoding (Hildebrand and Vogel, 2008).

Numerous attempts have been made to tackle the confidence estimation problem. The work of Blatz et al. (2004) is perhaps the best known study of sentence and word level features and their impact on translation error prediction. Along this line of research, improvements can be obtained by incorporating more features as shown in (Quirk, 2004; Sanchis et al., 2007; Raybaud et al., 2009; Specia et al., 2009). Soricut and Echihabi (2010) developed regression models which are used to predict the expected BLEU score of a given translation hypothesis. Improvement also can be obtained by using target part-of-speech and null dependency link in a MaxEnt classifier (Xiong et al., 2010). Ueffing and Ney (2007) introduced word posterior probabilities (WPP) features and applied them in the n-best list reranking. From the usability point of view, back-translation is a tool to help users to assess the accuracy level of MT output (Bach et al., 2007). Literally, it translates backward the MT output into the source language to see whether the output of backward translation matches the original source sentence.

However, previous studies had a few shortcomings. First, source-side features were not extensively investigated. Blatz et al.(2004) only investigated source n-gram frequency statistics and source language model features, while other work mainly focused on target side features. Second, previous work attempted to incorporate more features but faced scalability issues, i.e., to train many features we need many training examples and to train discriminatively we need to search through all possible translations of each training example. Another issue of previous work was that they are all trained with BLEU/TER score computing against the translation references which is different from predicting the human-targeted translation edit rate (HTER) which is crucial in post-editing applications (Snover et al., 2006; Papineni et al., 2002). Finally, the back-translation approach faces a serious issue when forward

* Work done during an internship at IBM T.J. Watson Research Center

and backward translation models are symmetric. In this case, back-translation will not be very informative to indicate forward translation quality.

In this paper, we predict error types of each word in the MT output with a confidence score, extend it to the sentence level, then apply it to n-best list reranking task to improve MT quality, and finally design a visualization prototype. We try to answer the following questions:

- Can we use a rich feature set such as source-side information, alignment context, and dependency structures to improve error prediction performance?
- Can we predict more translation error types i.e substitution, insertion, deletion and shift?
- How good do our prediction methods correlate with human correction?
- Do confidence measures help the MT system to select a better translation?
- How confidence score can be presented to improve end-user perception?

In Section 2, we describe the models and training method for the classifier. We describe novel features including source-side, alignment context, and dependency structures in Section 3. Experimental results and analysis are reported in Section 4. Section 5 and 6 present applications of confidence scores.

2 Confidence Measure Model

2.1 Problem setting

Confidence estimation can be viewed as a sequential labelling task in which the word sequence is MT output and word labels can be $Bad/Good$ or $Insertion/Substitution/Shift/Good$. We first estimate each individual word confidence and extend it to the whole sentence. Arabic text is fed into an Arabic-English SMT system and the English translation outputs are corrected by humans in two phases. In phase one, a bilingual speaker corrects the MT system translation output. In phase two, another bilingual speaker does quality checking for the correction done in phase one. If bad corrections were spotted, they correct them again. In this paper we use the final correction data from phase two as the reference thus HTER can be used as an evaluation metric. We have 75 thousand sentences with 2.4 million words in total from the human correction process described above.

We obtain training labels for each word by performing TER alignment between MT output and the phase-two human correction. From TER alignments we observed that out of total errors are 48% substitution, 28% deletion, 13% shift, and 11% insertion errors. Based on the alignment, each word produced by the MT system has a label: good, insertion, substitution and shift. Since a deletion error occurs when it only appears in the reference translation, not in the MT output, our model will not predict deletion errors in the MT output.

2.2 Word-level model

In our problem, a training instance is a word from MT output, and its label when the MT sentence is aligned with the human correction. Given a training instance x, y is the true label of x; f stands for its feature vector $f(x, y)$; and w is feature weight vector. We define a feature-rich classifier $score(x, y)$ as follow

$$score(x, y) = w.f(x, y) \tag{1}$$

To obtain the label, we choose the class with the highest score as the predicted label for that data instance. To learn optimized weights, we use the Margin Infused Relaxed Algorithm or MIRA (Crammer and Singer, 2003; McDonald et al., 2005) which is an online learner closely related to both the support vector machine and perceptron learning framework. MIRA has been shown to provide state-of-the-art performance for sequential labelling task (Rozenfeld et al., 2006), and is also able to provide an efficient mechanism to train and optimize MT systems with lots of features (Watanabe et al., 2007; Chiang et al., 2009). In general, weights are updated at each step time t according to the following rule:

$$\begin{aligned} & w_{t+1} = \arg\min_{w_{t+1}} ||w_{t+1} - w_t|| \\ \text{s.t.} \quad & score(x, y) \geq score(x, y') + L(y, y') \end{aligned} \tag{2}$$

where $L(y, y')$ is a measure of the loss of using y' instead of the true label y. In this problem $L(y, y')$ is 0-1 loss function. More specifically, for each instance x_i in the training data at a time t we find the label with the highest score:

$$y' = \arg\max_{y} score(x_i, y) \tag{3}$$

the weight vector is updated as follow

$$w_{t+1} = w_t + \tau(f(x_i, y) - f(x_i, y')) \tag{4}$$

τ can be interpreted as a step size; when τ is a large number we want to update our weights aggressively, otherwise weights are updated conservatively.

$$\begin{aligned} & \tau = \max(0, \alpha) \\ & \alpha = \min\left\{C, \frac{L(y,y') - (score(x_i,y) - score(x_i,y'))}{||f(x_i,y) - f(x_i,y')||_2^2}\right\} \end{aligned} \tag{5}$$

where C is a positive constant used to cap the maximum possible value of τ. In practice, a cut-off threshold n is the parameter which decides the number of features kept (whose occurrence is at least n) during training. Note that MIRA is sensitive to constant C, the cut-off feature threshold n, and the number of iterations. The final weight is typically normalized by the number of training iterations and the number of training instances. These parameters are tuned on a development set.

2.3 Sentence-level model

Given the feature sets and optimized weights, we use the Viterbi algorithm to find the best label sequence. To estimate the confidence of a sentence S we rely on the information from the forward-backward inference. One approach is to directly use the conditional probabilities of the whole sequence. However, this quantity is the confidence measure for the label sequence predicted by the classifier and it does not represent the goodness of the whole MT output. Another more appropriated method is to use the marginal probability of $Good$ label which can be defined as follow:

$$p(y_i = Good|S) = \frac{\alpha(y_i|S)\beta(y_i|S)}{\sum_j \alpha(y_j|S)\beta(y_j|S)} \quad (6)$$

$p(y_i = Good|S)$ is the marginal probability of label $Good$ at position i given the MT output sentence S. $\alpha(y_i|S)$ and $\beta(y_i|S)$ are forward and backward values. Our confidence estimation for a sentence S of k words is defined as follow

$$goodness(S) = \frac{\sum_{i=1}^{k} p(y_i = Good|S)}{k} \quad (7)$$

$goodness(S)$ is ranging between 0 and 1, where 0 is equivalent to an absolutely wrong translation and 1 is a perfect translation. Essentially, $goodness(S)$ is the arithmetic mean which represents the goodness of translation per word in the whole sentence.

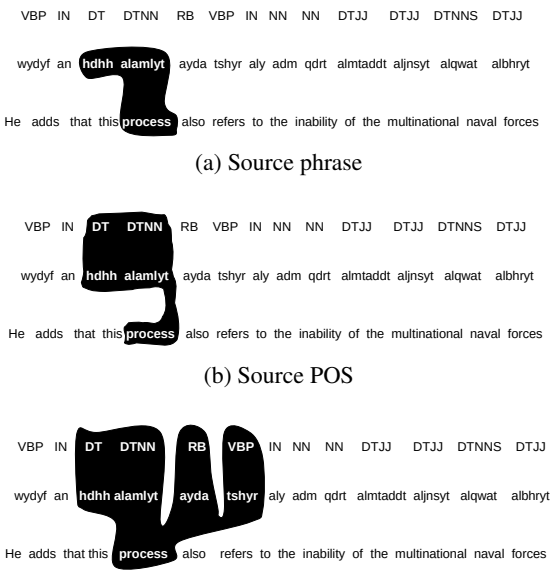
3 Confidence Measure Features

Features are generated from feature types: abstract templates from which specific features are instantiated. Features sets are often parameterized in various ways. In this section, we describe three new feature sets introduced on top of our baseline classifier which has WPP and target POS features (Ueffing and Ney, 2007; Xiong et al., 2010).

3.1 Source-side features

From MT decoder log, we can track which source phrases generate target phrases. Furthermore, one can infer the alignment between source and target words within the phrase pair using simple aligners such as IBM Model-1 alignment.

Source phrase features: These features are designed to capture the likelihood that source phrase and target word co-occur with a given error label. The intuition behind them is that if a large percentage of the source phrase and target have often been seen together with the same label, then the produced target word should have this label in the future. Figure 1a illustrates this feature template where the first line is source POS tags, the second line is the Buckwalter romanized source Arabic sequence, and the third line is MT output. The source phrase feature is defined as follow



(a) Source phrase

(b) Source POS

(c) Source POS and phrase in right context

Figure 1: Source-side features.

$$f_{102}(process) = \begin{cases} 1 & \text{if source-phrase="}\textit{hdhh alamlyt}\text{"} \\ 0 & \text{otherwise} \end{cases}$$

Source POS: Source phrase features might be susceptible to sparseness issues. We can generalize source phrases based on their POS tags to reduce the number of parameters. For example, the example in Figure 1a is generalized as in Figure 1b and we have the following feature:

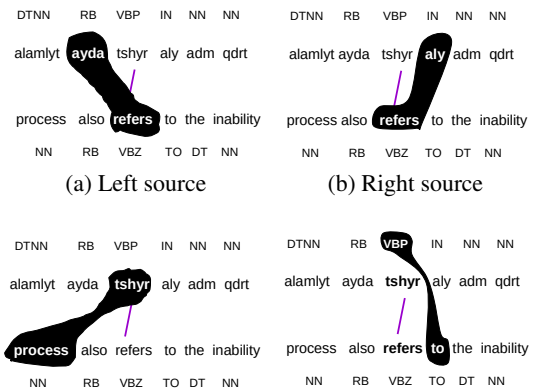
$$f_{103}(process) = \begin{cases} 1 & \text{if source-POS="}\textit{ DT DTNN }\text{"} \\ 0 & \text{otherwise} \end{cases}$$

Source POS and phrase context features: This feature set allows us to look at the surrounding context of the source phrase. For example, in Figure 1c we have "*hdhh alamlyt*" generates "*process*". We also have other information such as on the right hand side the next two phrases are "*ayda*" and "*tshyr*" or the sequence of source target POS on the right hand side is "*RB VBP*". An example of this type of feature is

$$f_{104}(process) = \begin{cases} 1 & \text{if source-POS-context="}\textit{ RB VBP }\text{"} \\ 0 & \text{otherwise} \end{cases}$$

3.2 Alignment context features

The IBM Model-1 feature performed relatively well in comparison with the WPP feature as shown by Blatz et al. (2004). In our work, we incorporate not only the IBM Model-1 feature but also the surrounding alignment context. The key intuition is that collocation is a reliable indicator for judging if a target word is generated by a particular source word (Huang, 2009). Moreover, the IBM Model-1 feature was already used in several steps of a translation system such as word alignment, phrase extraction and scoring. Also the impact of



(a) Left source (b) Right source

(c) Left target (d) Source POS & right target

Figure 2: Alignment context features.

this feature alone might fade away when the MT system is scaled up.

We obtain word-to-word alignments by applying IBM Model-1 to bilingual phrase pairs that generated the MT output. The IBM Model-1 assumes one target word can only be aligned to one source word. Therefore, given a target word we can always identify which source word it is aligned to.

Source alignment context feature: We anchor the target word and derive context features surrounding its source word. For example, in Figure 2a and 2b we have an alignment between "*tshyr*" and "*refers*" The source contexts "*tshyr*" with a window of one word are "*ayda*" to the left and "*aly*" to the right.

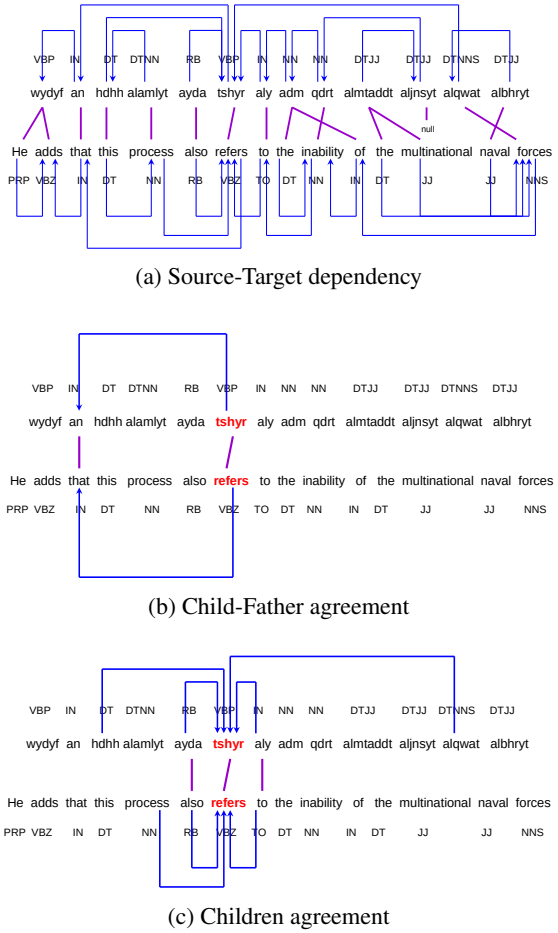
Target alignment context feature: Similar to source alignment context features, we anchor the source word and derive context features surrounding the aligned target word. Figure 2c shows a left target context feature of word "*refers*". Our features are derived from a window of four words.

Combining alignment context with POS tags: Instead of using lexical context we have features to look at source and target POS alignment context. For instance, the feature in Figure 2d is

$$f_{141}(refers) = \begin{cases} 1 & \text{if source-POS = "}VBP\text{"} \\ & \text{and target-context = "}to\text{"} \\ 0 & \text{otherwise} \end{cases}$$

3.3 Source and target dependency structure features

The contextual and source information in the previous sections only take into account surface structures of source and target sentences. Meanwhile, dependency structures have been extensively used in various translation systems (Shen et al., 2008; Ma et al., 2008; Bach et al., 2009). The adoption of dependency



(a) Source-Target dependency

(b) Child-Father agreement

(c) Children agreement

Figure 3: Dependency structures features.

structures might enable the classifier to utilize deep structures to predict translation errors. Source and target structures are unlikely to be isomorphic as shown in Figure 3a. However, we expect some high-level linguistic structures are likely to transfer across certain language pairs. For example, prepositional phrases (PP) in Arabic and English are similar in a sense that PPs generally appear at the end of the sentence (after all the verbal arguments) and to a lesser extent at its beginning (Habash and Hu, 2009). We use the Stanford parser to obtain dependency trees and POS tags (Marneffe et al., 2006).

Child-Father agreement: The motivation is to take advantage of the long distance dependency relations between source and target words. Given an alignment between a source word s_i and a target word t_j. A child-father agreement exists when s_k is aligned to t_l, where s_k and t_l are father of s_i and t_j in source and target dependency trees, respectively. Figure 3b illustrates that "*tshyr*" and "*refers*" have a child-father agreement. To verify our intuition, we analysed 243K words of manual aligned Arabic-English bitext. We observed 29.2% words having child-father agreements. In term of structure types, we found 27.2% of copula verb

and 30.2% prepositional structures, including object of a preposition, prepositional modifier, and prepositional complement, are having child-father agreements.

Children agreement: In the child-father agreement feature we look up in the dependency tree, however, we also can look down to the dependency tree with a similar motivation. Essentially, given an alignment between a source word s_i and a target word t_j, how many children of s_i and t_j are aligned together? For example, "*tshyr*" and "*refers*" have 2 aligned children which are "*ayda-also*" and "*aly-to*" as shown in Figure 3c.

4 Experiments

4.1 Arabic-English translation system

The SMT engine is a phrase-based system similar to the description in (Tillmann, 2006), where various features are combined within a log-linear framework. These features include source-to-target phrase translation score, source-to-target and target-to-source word-to-word translation scores, language model score, distortion model scores and word count. The training data for these features are 7M Arabic-English sentence pairs, mostly newswire and UN corpora released by LDC. The parallel sentences have word alignment automatically generated with HMM and MaxEnt word aligner (Ge, 2004; Ittycheriah and Roukos, 2005). Bilingual phrase translations are extracted from these word-aligned parallel corpora. The language model is a 5-gram model trained on roughly 3.5 billion English words.

Our training data contains 72k sentences Arabic-English machine translation with human corrections which include of 2.2M words in newswire and weblog domains. We have a development set of 2,707 sentences, 80K words (dev); an unseen test set of 2,707 sentences, 79K words (test). Feature selection and parameter tuning has been done on the development set in which we experimented values of C, n and iterations in range of [0.5:10], [1:5], and [50:200] respectively. The final MIRA classifier was trained by using pocket crf toolkit[1] with 100 iterations, hyper-parameter C was 5 and cut-off feature threshold n was 1.

We use precision (P), recall (R) and F-score (F) to evaluate the classifier performance and they are computed as follow:

$$\begin{aligned} P &= \frac{\text{the number of correctly tagged labels}}{\text{the number of tagged labels}} \\ R &= \frac{\text{the number of correctly tagged labels}}{\text{the number of reference labels}} \\ F &= \frac{2*P*R}{P+R} \end{aligned} \quad (8)$$

4.2 Contribution of feature sets

We designed our experiments to show the impact of each feature separately as well as their cumulative impact. We trained two types of classifiers to predict the error type of each word in MT output, namely Good/Bad with a binary classifier and Good/Insertion/Substitution/Shift with a 4-class classifier. Each classifier is trained with different feature sets as follow:

- WPP: we reimplemented WPP calculation based on n-best lists as described in (Ueffing and Ney, 2007).
- WPP + target POS: only WPP and target POS features are used. This is a similar feature set used by Xiong et al. (2010).
- Our features: the classifier has source side, alignment context, and dependency structure features; WPP and target POS features are excluded.
- WPP + our features: adding our features on top of WPP.
- WPP + target POS + our features: using all features.

	binary		4-class	
	dev	test	dev	test
WPP	69.3	68.7	64.4	63.7
+ source side	72.1	**71.6**	66.2	**65.7**
+ alignment context	71.4	70.9	65.7	65.3
+ dependency structures	69.9	69.5	64.9	64.3
WPP+ target POS	69.6	69.1	64.4	63.9
+ source side	72.3	**71.8**	66.3	**65.8**
+ alignment context	71.9	71.2	66	65.6
+ dependency structures	70.4	70	65.1	64.4

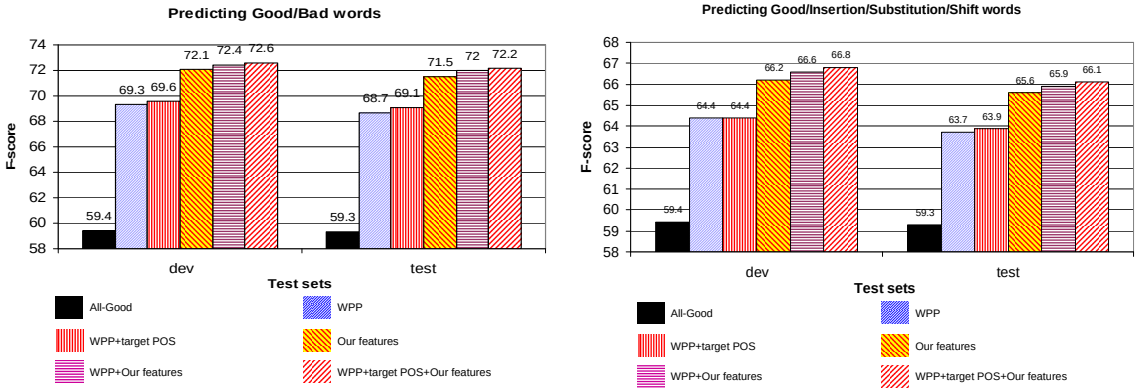
Table 1: Contribution of different feature sets measure in F-score.

To evaluate the effectiveness of each feature set, we apply them on two different baseline systems: using WPP and WPP+target POS, respectively. We augment each baseline with our feature sets separately. Table 1 shows the contribution in F-score of our proposed feature sets. Improvements are consistently obtained when combining the proposed features with baseline features. Experimental results also indicate that source-side information, alignment context and dependency structures have unique and effective levers to improve the classifier performance. Among the three proposed feature sets, we observe the source side information contributes the most gain, which is followed by the alignment context and dependency structure features.

4.3 Performance of classifiers

We trained several classifiers with our proposed feature sets as well as baseline features. We compare their performances, including a naive baseline All-Good classifier, in which all words in the MT output are labelled as good translations. Figure 4 shows the performance

[1] http://pocket-crf-1.sourceforge.net/



(a) Binary (b) 4-class

Figure 4: Performance of binary and 4-class classifiers trained with different feature sets on the development and unseen test sets.

of different classifiers trained with different feature sets on development and unseen test sets. On the unseen test set our proposed features outperform WPP and target POS features by 2.8 and 2.4 absolute F-score respectively. Improvements of our features are consistent in development and unseen sets as well as in binary and 4-class classifiers. We reach the best performance by combining our proposed features with WPP and target POS features. Experiments indicate that the gaps in F-score between our best system with the naive All-Good system is 12.9 and 6.8 in binary and 4-class cases, respectively. Table 2 presents precision, recall, and F-score of individual class of the best binary and 4-class classifiers. It shows that *Good* label is better predicted than other labels, meanwhile, $Substitution$ is generally easier to predict than $Insertion$ and $Shift$.

	Label	P	R	F
Binary	Good	74.7	80.6	77.5
	Bad	68	60.1	63.8
4-class	Good	70.8	87	78.1
	Insertion	37.5	16.9	23.3
	Substitution	57.8	44.9	50.5
	Shift	35.2	14.1	20.1

Table 2: Detailed performance in precision, recall and F-score of binary and 4-class classifiers with WPP+target POS+Our features on the unseen test set.

4.4 Correlation between Goodness and HTER

We estimate sentence level confidence score based on Equation 7. Figure 5 illustrates the correlation between our proposed $goodness$ sentence level confidence score and the human-targeted translation edit rate (HTER). The Pearson correlation between $goodness$ and HTER is 0.6, while the correlation of WPP and HTER is 0.52. This experiment shows that

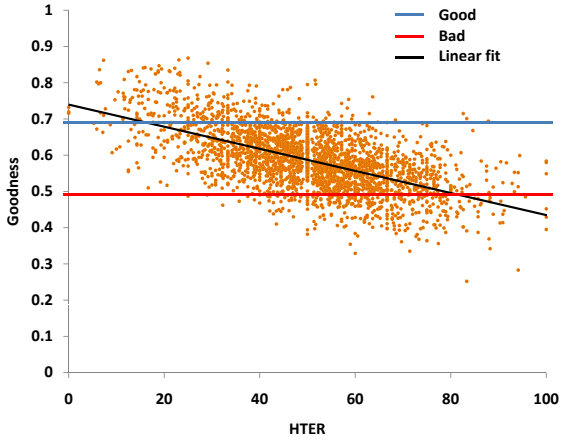


Figure 5: Correlation between Goodness and HTER.

$goodness$ has a large correlation with HTER. The black bar is the linear regression line. Blue and red bars are thresholds used to visualize good and bad sentences respectively. We also experimented $goodness$ computation in Equation 7 using geometric mean and harmonic mean; their Pearson correlation values are 0.5 and 0.35 respectively.

5 Improving MT quality with N-best list reranking

Experiments reporting in Section 4 indicate that the proposed confidence measure has a high correlation with HTER. However, it is not very clear if the core MT system can benefit from confidence measure by providing better translations. To investigate this question we present experimental results for the n-best list reranking task.

The MT system generates top n hypotheses and for each hypothesis we compute sentence-level confidence scores. The best candidate is the hypothesis with highest confidence score. Table 3 shows the performance of

	Dev		Test	
	TER	BLEU	TER	BLEU
Baseline	49.9	31.0	50.2	30.6
2-best	49.5	31.4	49.9	30.8
5-best	**49.2**	**31.4**	**49.6**	**30.8**
10-best	49.2	31.2	49.5	30.8
20-best	49.1	31.0	49.3	30.7
30-best	49.0	31.0	49.3	30.6
40-best	49.0	31.0	49.4	30.5
50-best	49.1	30.9	49.4	30.5
100-best	49.0	30.9	49.3	30.5

Table 3: Reranking performance with *goodness* score.

reranking systems using *goodness* scores from our best classifier in various n-best sizes. We obtained 0.7 TER reduction and 0.4 BLEU point improvement on the development set with a 5-best list. On the unseen test, we obtained 0.6 TER reduction and 0.2 BLEU point improvement. Although, the improvement of BLEU score is not obvious, TER reductions are consistent in both development and unseen sets. Figure 6 shows the improvement of reranking with *goodness* score. Besides, the figure illustrates the upper and lower bound performances with TER metric in which the lower bound is our baseline system and the upper bound is the best hypothesis in a given n-best list. Oracle scores of each n-best list are computed by choosing the translation candidate with lowest TER score.

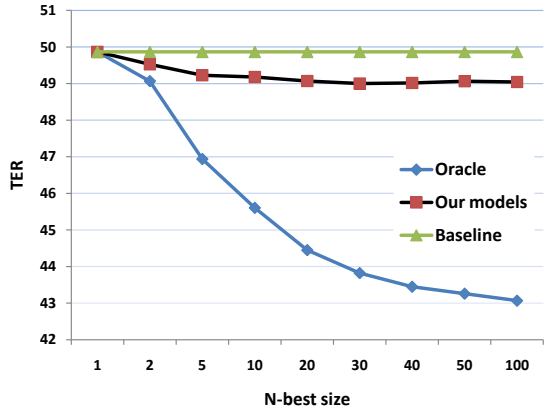


Figure 6: A comparison between reranking and oracle scores with different n-best size in TER metric on the development set.

6 Visualizing translation errors

Besides the application of confidence score in the n-best list reranking task, we propose a method to visualize translation error using confidence scores. Our purpose is to visualize word and sentence-level confidence scores with the following objectives 1) easy for spotting translations errors; 2) simple and intuitive; and 3) helpful for post-editing productivity. We define three categories of translation quality (good/bad/decent) on both word and sentence level. On word level, the marginal probability of good label is used to visualize translation errors as follow:

$$L_i = \begin{cases} good & \text{if } p(y_i = Good|S) \geq 0.8 \\ bad & \text{if } p(y_i = Good|S) \leq 0.45 \\ decent & \text{otherwise} \end{cases}$$

On sentence level, the goodness score is used as follow:

$$L_S = \begin{cases} good & \text{if } goodness(S) \geq 0.7 \\ bad & \text{if } goodness(S) \leq 0.5 \\ decent & \text{otherwise} \end{cases}$$

	Choices	Intention
Font size	big	bad
	small	good
	medium	decent
Colors	red	bad
	black	good
	orange	decent

Table 4: Choices of layout

Different font sizes and colors are used to catch the attention of post-editors whenever translation errors are likely to appear as shown in Table 4. Colors are applied on word level, while font size is applied on both word and sentence level. The idea of using font size and colour to visualize translation confidence is similar to the idea of using tag/word cloud to describe the content of websites[2]. The reason we are using big font size and red color is to attract post-editors' attention and help them find translation errors quickly. Figure 7 shows an example of visualizing confidence scores by font size and colours. It shows that "*not to deprive yourself*", displayed in big font and red color, is likely to be bad translations. Meanwhile, other words, such as "*you*", "*different*", "*from*", and "*assimilation*", displayed in small font and black color, are likely to be good translation. Medium font and orange color words are decent translations.

7 Conclusions

In this paper we proposed a method to predict confidence scores for machine translated words and sentences based on a feature-rich classifier using linguistic and context features. Our major contributions are three novel feature sets including source side information, alignment context, and dependency structures. Experimental results show that by combining the source side information, alignment context, and dependency structure features with word posterior probability and target POS context (Ueffing & Ney 2007; Xiong et al.,

[2]http://en.wikipedia.org/wiki/Tag_cloud

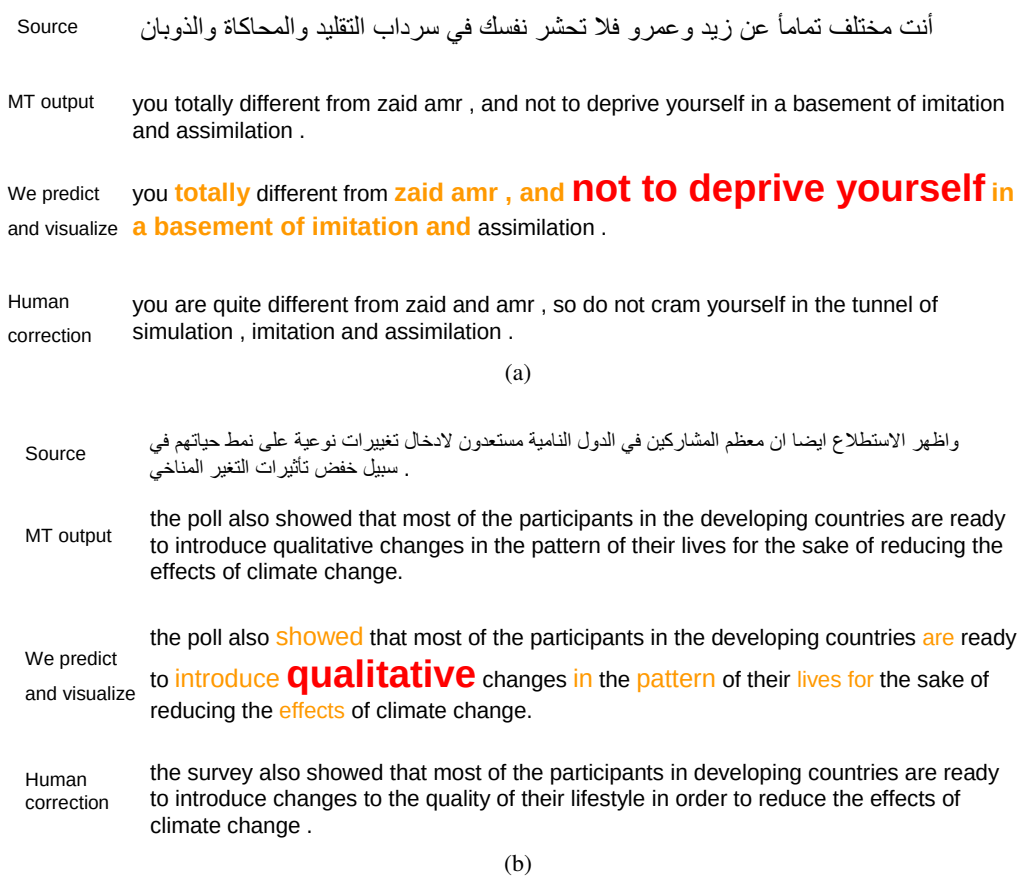


Figure 7: MT errors visualization based on confidence scores.

2010), the MT error prediction accuracy is increased from 69.1 to 72.2 in F-score. Our framework is able to predict error types namely insertion, substitution and shift. The Pearson correlation with human judgement increases from 0.52 to 0.6. Furthermore, we show that the proposed confidence scores can help the MT system to select better translations and as a result improvements between 0.4 and 0.9 TER reduction are obtained. Finally, we demonstrate a prototype to visualize translation errors.

This work can be expanded in several directions. First, we plan to apply confidence estimation to perform a second-pass constraint decoding. After the first pass decoding, our confidence estimation model can label which word is likely to be correctly translated. The second-pass decoding utilizes the confidence information to constrain the search space and hopefully can find a better hypothesis than in the first pass. This idea is very similar to the multi-pass decoding strategy employed by speech recognition engines. Moreover, we also intend to perform a user study on our visualization prototype to see if it increases the productivity of post-editors.

Acknowledgements

We would like to thank Christoph Tillmann and the IBM machine translation team for their supports. Also, we would like to thank anonymous reviewers, Qin Gao, Joy Zhang, and Stephan Vogel for their helpful comments.

References

Nguyen Bach, Matthias Eck, Paisarn Charoenpornsawat, Thilo Khler, Sebastian Stker, ThuyLinh Nguyen, Roger Hsiao, Alex Waibel, Stephan Vogel, Tanja Schultz, and Alan Black. 2007. The CMU TransTac 2007 Eyes-free and Hands-free Two-way Speech-to-Speech Translation System. In *Proceedings of the IWSLT'07*, Trento, Italy.

Nguyen Bach, Qin Gao, and Stephan Vogel. 2009. Source-side dependency tree reordering models with subtree movements and constraints. In *Proceedings of the MTSummit-XII*, Ottawa, Canada, August. International Association for Machine Translation.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *The JHU Workshop Final Report*, Baltimore, Maryland, USA, April.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of HLT-ACL*, pages 218–226, Boulder, Colorado, June. Association for Computational Linguistics.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Niyu Ge. 2004. Max-posterior HMM alignment for machine translation. In *Presentation given at DARPA/TIDES NIST MT Evaluation workshop*.

Nizar Habash and Jun Hu. 2009. Improving arabic-chinese statistical machine translation using english as pivot language. In *Proceedings of the 4th Workshop on Statistical Machine Translation*, pages 173–181, Morristown, NJ, USA. Association for Computational Linguistics.

Almut Silja Hildebrand and Stephan Vogel. 2008. Combination of machine translation systems via hypothesis selection from combined n-best lists. In *Proceedings of the 8th Conference of the AMTA*, pages 254–261, Waikiki, Hawaii, October.

Fei Huang. 2009. Confidence measure for word alignment. In *Proceedings of the ACL-IJCNLP '09*, pages 932–940, Morristown, NJ, USA. Association for Computational Linguistics.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of the HTL-EMNLP'05*, pages 89–96, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL'07*, pages 177–180, Prague, Czech Republic, June.

Yanjun Ma, Sylwia Ozdowska, Yanli Sun, and Andy Way. 2008. Improving word alignment using syntactic dependencies. In *Proceedings of the ACL-08: HLT SSST-2*, pages 69–77, Columbus, OH.

Marie-Catherine Marneffe, Bill MacCartney, and Christopher Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC'06*, Genoa, Italy.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Flexible text segmentation with structured multilabel classification. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 987–994, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL'02*, pages 311–318, Philadelphia, PA, July.

Chris Quirk. 2004. Training a sentence-level machine translation confidence measure. In *Proceedings of the 4th LREC*.

Sylvain Raybaud, Caroline Lavecchia, David Langlois, and Kamel Smaili. 2009. Error detection for statistical machine translation using linguistic features. In *Proceedings of the 13th EAMT*, Barcelona, Spain, May.

Binyamin Rozenfeld, Ronen Feldman, and Moshe Fresko. 2006. A systematic cross-comparison of sequence classifiers. In *Proceedings of the SDM*, pages 563–567, Bethesda, MD, USA, April.

Alberto Sanchis, Alfons Juan, and Enrique Vidal. 2007. Estimation of confidence measures for machine translation. In *Proceedings of the MT Summit XI*, Copenhagen, Denmark.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA'06*, pages 223–231, August.

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

Lucia Specia, Zhuoran Wang, Marco Turchi, John Shawe-Taylor, and Craig Saunders. 2009. Improving the confidence of machine translation quality estimates. In *Proceedings of the MT Summit XII*, Ottawa, Canada.

Christoph Tillmann. 2006. Efficient dynamic programming search algorithms for phrase-based SMT. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 9–16, Morristown, NJ, USA. Association for Computational Linguistics.

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

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the EMNLP-CoNLL*, pages 764–773, Prague, Czech Republic, June. Association for Computational Linguistics.

Deyi Xiong, Min Zhang, and Haizhou Li. 2010. Error detection for statistical machine translation using linguistic features. In *Proceedings of the 48th ACL*, pages 604–611, Uppsala, Sweden, July. Association for Computational Linguistics.