Integrating Lexical, Syntactic and System-based Features to Improve Word Confidence Estimation in SMT

Luong Ngoc Quang
Laboratoire LIG, GETALP, Grenoble, France
Ngoc-Quang.Luong@imag.fr

RESUME

Intégration de paramètres lexicaux, syntaxiques et issus du système de traduction automatique pour améliorer l'estimation des mesures de confiance au niveau des mots

L'estimation des mesures de confiance (MC) au niveau des mots consiste à prédire leur exactitude dans la phrase cible générée par un système de traduction automatique. Ceci permet d'estimer la fiabilité d'une sortie de traduction et de filtrer les segments trop mal traduits pour une post-édition. Nous étudions l'impact sur le calcul des MC de différents paramètres : lexicaux, syntaxiques et issus du système de traduction. Nous présentons la méthode permettant de labelliser automatiquement nos corpus (mot correct ou incorrect), puis le classifieur à base de champs aléatoires conditionnels utilisé pour intégrer les différents paramètres et proposer une classification appropriée des mots. Nous avons effectué des expériences préliminaires, avec l'ensemble des paramètres, où nous mesurons la précision, le rappel et la F-mesure. Finalement nous comparons les résultats avec notre système de référence. Nous obtenons de bons résultats pour la classification des mots considérés comme corrects (F-mesure : 86.7%), et encourageants pour ceux estimés comme mal traduits (F-mesure : 36,8%).

ABSTRACT

Confidence Estimation at word level is the task of predicting the correct and incorrect words in the target sentence generated by a MT system. It helps to conclude the reliability of a given translation as well as to filter out sentences that are not good enough for post-editing. This paper investigates various types of features to circumvent this issue, including lexical, syntactic and system-based features. A method to set training label for each word in the hypothesis is also presented. A classifier based on conditional random fields (CRF) is employed to integrate features and determine the word's appropriate label. We conducted our preliminary experiment with all features, tracked precision, recall and F-score and we compared with our baseline system. Experimental results of the full combination of all features yield the very encouraging precision, recall and F-score for Good label (F-score: 86.7%), and acceptable scores for Bad label (F-score: 36.8%).

MOTS-CLES : Système de traduction automatique, mesure de confiance, estimation de la confiance, champs aléatoires conditionnels
KEYWORDS : Machine translation, confidence measure, confidence estimation, conditional random fields

Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 3: RECITAL, pages 43–56,
Grenoble, 4 au 8 juin 2012. ©2012 ATALA & AFCP

1 Introduction

Statistical Machine Translation Systems in recent years have marked impressive breakthroughs with numerous fruitful achievements, as they produced more and more user-acceptable outputs. Nevertheless users have to face with some big questions that still remain open: are these translations ready to be published or some post-edit operations will be needed? Are they worth to be corrected or the re-translation from scratch is less time-consuming? It is undoubtedly that building a method which is capable of pointing out the correct parts as well as detecting the translation errors in each MT hypothesis is crucial to tackle these above issues. If we limit the concept "parts" to "words", the problem is called Word-level Confidence Estimation.

The objective of Word-level Confidence Estimation is to judge each word in the hypothesis as correct or incorrect by tagging it with an appropriate label. A classifier which has been trained beforehand by a feature set calculates the confidence score for MT output word, and then compares it with a threshold. All words with scores that exceed this threshold are categorized in the Good label set; the rest will belong to Bad label set.

Contributions of Confidence Estimation for the other aspects of Machine Translation are incontestable. Firstly, it assists the post-editors to quickly and intuitively identify the translation errors, and then they can determine whether to correct the sentence or re-translate it from scratch. This support gains lots of both post-editing time and efforts. Second, confidence score assigned to words is a potential clue to re-rank the MT hypothesis, thus improve its translation quality. Last but not least, it can be used by the translator in an interactive scenario (Gandrabur and Foster, 2003).

This article presents the integration of various types of features into CRF model to forecast the label for each word in the MT hypothesis. We organize the remaining parts as follow: in Section 2, we briefly review some previous researches related to confidence estimation at word level. The concept of CRF model, which we use to train our feature set will be introduced in Section 3. Section 4 details various system-based, lexical and syntactic features exploited for the classifier construction. Section 5 lists our settings to prepare for the preliminary experiments. The preliminary experiments together with their results are reported in Section 6. Lastly, section 7 concludes the paper and points out some perspectives.

2 Previous Work Review

To cope with Word-level Confidence Estimation problem, various approaches have been proposed, and most of them concentrate on the two major issues: which type of features and their combinations are efficient? And which classifier is the most suitable for training the feature sets?

In one of the earliest as well as most well-known work in this area, (Blatz et al., 2003) combine a considerable number of features by applying neural network and naïve Bayes learning algorithms. Among these features, the N-best lists based features, especially Word Posterior Probability (henceforth WPP) proposed in (Ueffing et al., 2003) have been shown to be one of the most effective system-based features by their experimental results. The combination of WPP (with 3 different proposed variants) and the IBM-Model 1 based features are also confirmed to overwhelm all the other single ones, including heuristic and semantic features in terms of performance in (Blatz et al., 2004). Using solely N-best list, (Sanchis et al., 2007) suggest 9 different features and then adopt a smoothed naïve Bayes classification model for training them.

(Ueffing and Ney, 2005) introduce a novel approach which explicitly explores the phrased-based translation model for detecting word errors. The phrase is considered as a contiguous sequence of words and is extracted from word-aligned bilingual training corpus for both source and target sides. The confidence value of each target word is then computed by summing over all phrase pairs in which the target part contains this word. Experimental results indicate that their method yielded an impressive reduction of the classification error rate compared to the state-of-the-art ones on the same language pairs employed.

(Xiong et al., 2010) integrate the POS of target word with another lexical feature named null dependency link and train them by MaxEnt classifier. In their results, the linguistic features sharply outperform word posterior probability feature in terms of F-score and classification error rate.

Unlike most of previous work, (Soricut and Echihabi, 2010) applied solely the external features of MT system with the hope that their classifier can deal with various MT approaches, from statistical-based to rule-based one. Given an MT output, the BLEU score is forecast due to the regression model they developed.

(Bach et al., 2011) study a method to calculate the confidence score for both generated target words and sentences relied on a feature-rich classifier. The features employed include source side information, alignment context, and dependency structure. The integration between them and Word posterior probability and POS context of target language helps to augment marginally in F-score as well as the Pearson correlation with human judgment.

Our work differs from previous researches at these main points: firstly, we investigate and integrate various types of features: system-based features extracted from the MT outputs (N-best lists with the score of the log-linear model as well as source and target language model features), together with lexical and syntactic features to see if this combination helps to improve the classifier's performance. All results observed will be reported in Section 6. Secondly, instead of using Levenshtein alignment or TER-p for generating the training label, we propose to use TERp-A thanks to some advantages which will be pointed out in Section 5. Thirdly, we apply the CRF model for integrating our predictor features as well as to classify words in the test set, which has been proven to avoid limitations of Markov models and stochastic grammars (Lafferty et al., 2001).

3 Conditional Random Fields Model for Confidence Estimation

CRF (Lafferty et al., 2001) is a framework for building probabilistic models for segmenting and labeling sequence data. The core idea of CRF can be summarized as follow: let $X = (x_1, x_2, ..., x_N)$ be the random variable over data sequence to be labeled, $Y = (y_1, y_2, ..., y_N)$ be the output sequence obtained after the labeling task. In our case, X ranges over words in the MT output, and Y represents the labels tagged for words. Each element y_i $(i=\overline{1,N})$ is assigned one value in the binary set $Y^N=\{Good, Bad\}$. The probability of sequence Y given X is written as:

$$p_\theta(Y \mid X)=\frac{1}{Z_\theta(X)}\exp\left\{\sum_{k=1}^{K}\theta_k F_k(X,Y)\right\} \tag{1}$$

where $$F_k(X,Y)=\sum_{t=1}^{T} f_k(y_{t-1},y_t,x_t) \tag{2}$$

$\{f_k\}(k=\overline{1,K})$ is a set of feature functions, $\{\theta_k\}(k=\overline{1,K})$ are the associated parameter values, and $Z_\theta(x)$ is a normalization factor, in which, the value is calculated by:

$$Z_\theta(x)=\sum_{y\in Y^N}\exp\left\{\sum_{k=1}^{K}\theta_k F_k(X,Y)\right\} \tag{3}$$

In order to estimate the conditional maximum likelihood given T independent sequences {X^i, Y^i} $(i=\overline{1,T})$ where X^i and Y^i contains N^i symbols, we have to minimize the negated conditional log-likelihood of the observations, with respect to θ:

$$\begin{aligned} l(\theta) &= -\sum_{i=1}^{T}\log p_\theta(Y_i \mid X_i) \\ &= \sum_{i=1}^{T}\left\{\log Z_\theta(X^i)-\sum_{k=1}^{K}\theta_k F_k(X^i,Y^i)\right\} \end{aligned} \tag{4}$$

The standard solution for this minimization is to apply an additional l^2 penalty term, determined by $\frac{\rho_2}{2}\|\theta\|_2^2$, where ρ_2 is a regularization parameter. The objective function is then a smooth convex function to be minimized over an unconstrained parameter space. Besides l^2, l^1 penalty calculated by $\rho_1\|\theta\|_1$ can also be exploited to perform the feature selection. It plays the role of controlling the amount of regularization as well as the number of extracted features. The combination of them helps to decrease the number of nonzero coefficients and to avoid the numerical problems which can appear in a huge dimensional parameter environment. The objective function corresponding to this combination will be $l(\theta)+\rho_1\|\theta\|_1+\frac{\rho_2}{2}\|\theta\|_2^2$.

Several optimization and regularization methods have been proposed to alleviate the parameter estimation issue. The most dominant algorithms among them are provided in WAPITI (Lavergne et al., 2010) – the CRF based labeling toolkit which we employed to combine our features, including: quasi-Newton (L-BFGS and OWL-QN), resilient propagation (R-PROP), stochastic gradient descent (SGD-L1), block-wise coordinate descent (BCD). We investigate the stochastic gradient descent to optimize our feature weights. In the labeling phase, we set the iterations for threshold from 0.3 to 1, with step of 0.025. In each loop, if the probability P("Good"|w) is greater than or equal this threshold, the corresponding word w will be tagged as "Good", and otherwise "Bad". This allows us to obtain a performance curve.

4 Exploitation of Various Kinds of Features

We explore three kinds of features, including:

4.1 System-based Features

They are the features extracted directly from our baseline SMT system based on Moses decoder options stated in Section 5.1, without the participation of any additional element. Based on the resource where features are found, they can be sub-categorized as following:

4.1.1 Target Side Features

We take into account the information in the MT output words, including:

- The word itself.
- The bi-gram sequences formed between current word and its precedence ($i-1/i$) or successor ($i/i+1$).
- The trigram sequences formed between current word and its two precedent and two following words (including: $i-2/i-1/i$; $i-1/i/i+1$; $i/i+1/i+2$).

4.1.2 Source Side Features

Using the alignment information between each target and source sentence, we can easily track the source words which the target word is aligned to. Unlike IBM Model-1 (Brown et al., 1993a) which supposes that each target word can be aligned to at most one source word; we process also the situation in which a phrase in the source sentence translates as a single word in the target sentence. To facilitate the alignment representation, we applied the BIO[1] format. In case of multiple target words aligned with one source word, the first word's alignment information will be prefixed with symbols "B-" (means Begin); and "I-" (means Inside) will be added at the beginning of alignment information for each of the remaining ones. With the target words which are not aligned with any source word, alignment information will be represented as O.

Target words (MT output)	Source aligned words		Target words (MT output)	Source aligned words
The	B-le		to	B-de
public	B-public		look	B-tourner
will	B-aura		again	B-à|nouveau
soon	B-bientôt		at	I-à
have	I-aura		its	B-son

[1] See more at: http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

the	B-I'		attention	B-attention
opportunity	B-occasion		.	B-.

TABLE 1 – Example of using BIO format to represent alignment information between source sentence and MT hypothesis.

Table 1 shows an example for this representation: since two target words "*will*" and "*have*" are aligned to "*aura*" in source sentence, the alignment information for them will be "*B-aura*" and "*I-aura*" respectively. In case a target word has multiple aligned source words (such as "*again*"), we separate these partners by symbol "|" after putting the prefix "B-" at the beginning.

4.1.3 Alignment Context Features

These features are proposed by (Bach et al., 2011) in regard with the intuition that collocation is a believable indicator for judging if a target word is generated by a particular source word. We also apply them in our experiments, containing:

- *Source alignment context features*: they are the patterns built from each target word and the surroundings of its source word. More precisely, we combine it with one word in the left (left source feature) or in the right (right source feature) of source word.
- *Target alignment context features*: similarly, we anchor the source word with all surroundings of the current target word. Since the window of size ± 2 is employed, it is obvious that 4 combinations will be generated.

4.1.4 Word Posterior Probability

As stated before, WPP has been proven to be one of the most prominent system-based features for confidence estimation. This is the likelihood of the word occurring in the target sentence, given the source sentence. Numerous knowledge sources have been proposed to calculate this value, such as word graphs, N-best lists, statistical word or phrase lexical. The key point here is to determine sentences in N-best lists that contain the word *e* under consideration in a fixed position *i*.

Let $p(f_1^J, e_1^I)$ be the joint probability of source sentence f_1^J and target sentence e_1^I. The word posterior probability of e occurring in position i is computed by aggregating probabilities of all sentences containing e in this position:

$$p_i(e \mid f_1^J) = \frac{p_i(e, f_1^J)}{\sum_{e'} p_i(e', f_1^J)} \quad (5)$$

where $$p_i(e, f_1^J) = \sum_{I, e_1^I} \theta(e_i, e).p(f_1^J, e_1^I) \quad (6)$$

Here $\theta(.,.)$ is the Kronecker function. The normalization in equation (5) is

$$\sum_{e'} p_i(e', f_1^J) = \sum_{I, e_1^I} p(f_1^J, e_1^I) = p(f_1^J) \quad (7)$$

In this work, we investigate the word graph that represents MT hypotheses (Ueffing, Och, and Ney 2002; Zens and Ney 2005). Thanks to this graph, the posterior probability of word e in position i can be calculated by summing up the probabilities of all paths that contains an edge annotated with e in position i of the target sentence. We perform this summation by applying the forward-backward algorithm (Jurafsky and Martin, 2000). This algorithm also determines the total probability mass needed for normalization, as shown in equation (7).

4.1.5 Target and Source Language Model Based Features

Applying SRILM toolkit (Stolcke, 2002) with the bilingual corpus, we build the 4-gram language model for both target and source side. These language models permit to identify the n-gram (length of the longest sequence created by the current token and its previous ones in the language model) of each word in MT output as well as in the source sentence. For example, with the current token w_i: if the sequence $w_{i-2}w_{i-1}w_i$ appears in the language model but the sequence $w_{i-3}w_{i-2}w_{i-1}w_i$ does not, the n-gram value for w_i will be 3. The value set for each word hence ranges from 0 to 4. Similarly, we extract the n-gram value for the source word aligned to w_i as one more feature.

4.2 Lexical Features

One of the most prominent lexical features that have been widely explored in Confidence Estimation researches is Word's Part-Of-Speech (POS). This tag is assigned to each word in a sentence due to its syntactic and morphological behaviors to indicate its lexical category. In our work, we chose TreeTagger[1] tool for POS annotation task in both source and target sides.

We implement these following lexical characteristics:

- POS of current target word.
- Sequence of POS of all source words which this target word is aligned to, represented in BIO format like alignment representation mentioned in Section 4.1.2.
- Besides using POS of each word in the target side as one lexical feature, we also observed a window of size n (n = 2 and n = 3) over the neighboring target positions and build the n-gram sequence for POS. More specifically, with n = 2 we get the POS sequences $i-1,i$ and $i,i+1$; with n = 3 we have 3 sequences: $i-2,i-1,i$; $i-1,i,i+1$and $i,i+1,i+2$.

4.3 Syntactic Features

Besides lexical features, the syntactic information of word in a sentence is also a potential clue for predicting its correctness. The intuition behind this is that if a word has grammatical relations with the others, it will be more likely to be correct than a word which has no relation. In order to obtain the links between words, we select the

[1] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

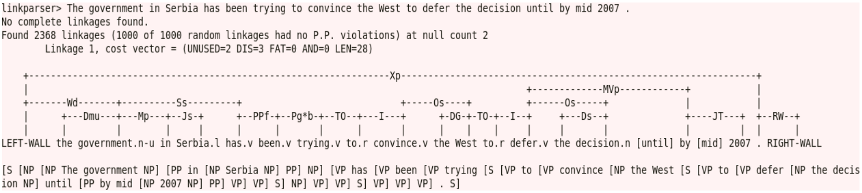
Link Grammar Parser[1] as our syntactic parser, allowing us to assign to each MT hypothesis a syntactic structure in which all pairs of words related together will be connected by a labeled link. In case of Link Grammar fails to find the full linkage for the whole sentence, it will skip each word one time until the sub-linkage for the remaining words has been successfully constructed. Based on this structure, we extract the following characteristics to build features:

- The Null Link property: does this word have link with the others or not?
- The total number of links this word has.

Another benefit yielded by Link Grammar Parser is the "constituent" tree (Penn tree-bank style phrase tree) to represent a sentence's grammatical structure (showing noun phrases, verb phrases, etc.). This constituent tree enables us to produce more syntactic features for word, including:

- Its constituent label.
- Its depth in the tree (or the distance between it and the tree root).

Figure 1 represents the syntactic structure as well as the constituent tree for a MT output: *"The government in Serbia has been trying to convince the West to defer the decision until by mid 2007."*.



```
linkparser> The government in Serbia has been trying to convince the West to defer the decision until by mid 2007 .
No complete linkages found.
Found 2368 linkages (1000 of 1000 random linkages had no P.P. violations) at null count 2
        Linkage 1, cost vector = (UNUSED=2 DIS=3 FAT=0 AND=0 LEN=28)

    +------------------------------------------------------------------Xp-----------------------------------------------------------------+
    |                                                                                           +------------MVp-------------+            |
    +-------Wd-------+----------Ss---------+                             +-----Os----+          +------Os-----+              |            |
    |      +---Dmu---+---Mp---+--Js-+      +--PPf-+--Pg*b-+--TO--+---I---+      +-DG-+-TO-+--I--+     +---Ds--+              +----JT---+  +--RW--+
    |      |         |        |     |      |      |       |      |       |      |    |    |     |     |       |              |         |  |      |
LEFT-WALL the government.n-u in Serbia.l has.v been.v trying.v to.r convince.v the West to.r defer.v the decision.n [until] by [mid] 2007 . RIGHT-WALL

[S [NP [NP The government NP] [PP in [NP Serbia NP] PP] NP] [VP has [VP been [VP trying [S [VP to [VP convince [NP the West [S [VP to [VP defer [NP the decis
ion NP] until [PP by mid [NP 2007 NP] PP] VP] VP] S] NP] VP] VP] S] VP] VP] VP] . S]
```

FIGURE 1 - Example of parsing result generated by Link Grammar

It is intuitive to observe in the graphic representation that the words in brackets (including *"until"* and *"mid"*) have no link with the others, meanwhile the remaining ones have. For instance, the word *"trying"* is connected with *"to"* by the link "TO" and with *"been"* by the link "Pg*b". Hence the "Null Link" and "Total Number of Links" for the word *"mid"* are true, 0 and for the word *"trying"* are false, 2 respectively. The figure also brings us the constituent label and the distance to the root of each word. In case of *"government"*, these values are NP and 2, respectively.

5 Experimental Settings

5.1 French – English SMT System Construction

Our baseline French – English SMT system was constructed using the Moses toolkit (Koehn et al., 2007). This toolkit contains all of necessary components to train the

[1] http://www.link.cs.cmu.edu/link/

translation model. In our work, we kept the Moses's default setting: log-linear model with 14 weighted feature functions. To train the translation model, we used the Europarl and News parallel corpora that are used for WMT[1] evaluation campaign in 2010 (total 1,638,440 sentences). Our target language model is a standard n-gram language model trained using the SRI language modeling toolkit (Stocke, 2002) on the news monolingual corpus (48,653,884 sentences). More details on this baseline system can be found in (Potet et al., 2010).

Besides this, in the decoder phase, we also called some extended options of Moses for tracking both source and target sides information which is mandatory to build our system-based features later. The most pivotal options are listed in the Table 2.

Option name	Function
-print-alignment-info-in-n-best	Display source-to-target and target-to-source word-to-word alignments into the N-best list.
-n-best-list FILE SIZE [distinct]	Generate an n-best file of up to SIZE distinct sentences into file FILE.

TABLE 2 – Moses options employed for tracking alignment information and N-best lists.

5.2 Corpus Preparation

We use our above SMT System to generate the translation hypothesis for 10,881 source sentences taken from several news corpora of the WMT evaluation campaign (from 2006 to 2010). A post-edition task was implemented by using a crowd sourcing platform: Amazon's Mechanical Turk (MTurk), which allows a "requester" to propose a paid or unpaid work and a "worker" to perform the proposed tasks. To avoid the huge gaps between the hypothesis and its post-edition since the correctors can paraphrase or reorder words to form the smoother translation, we highly recommended them to keep the number of edit operations as low as possible, but still ensure the accuracy of this translation with the French sentence. A sub-set (containing 311 sentences) of these collected post-editions was evaluated by a former professional post-editor. Testing result showed that 87.1% of post-editions improve the hypothesis. Detailed description for the corpus construction can be found in (Potet et al., 2012). Finally we extracted randomly 10,000 sentences triples (including source sentence, translation hypothesis and post-edited hypothesis) to form the training set, and keep the remaining 881 sentence triples for the test set.

5.3 Word Label Setting Using TERp-A

In order to obtain the training labels for each word in the MT outputs, previous works have made several attempts. (Xiong et al., 2010) exploited the Levenshtein alignment between the hypothesis and its best reference translation for classifying a word as correct or incorrect. In another method, the Translation Error Rate (TER) alignment

[1] http://www.statmt.org/wmt10/

was performed by (Bach et al., 2011), yielding one of the following labels for each word: good, insertion, substitution and shift. Nevertheless these above studies expressed some drawbacks. The hypothesis and its reference may differ in word order even when they have close meaning. Levenshtein alignment may not be able to align shifted words; hence it leads the inaccurate classification results. TER can be considered as a better alignment tool as it overcomes the first approach by enabling the block movement of words in the MT hypothesis and treating it equally with the other edit operations in term of cost edit, however the exact matches quality still remains limited since it lacks some crucial linguistic edit operations, and its edit costs are not well correlated with various type of human judgments. In order to propose a better word label tagging, we utilize the TER-Plus[1] (or TERp) toolkit. TERp is an extension of TER, not only inheriting the success of this evaluation metric and alignment tool, but also eliminating its shortcomings by taking into account the linguistic edit operations, such as Stem matches, Synonyms matches and Phrase Substitutions besides the TER's conventional ones (Exact match, Insertion, Deletion, Substitution and Shift). These additions allow us to avoid categorizing the hypothesis word as Insertion or Substitution in case that it shares same stem, or belongs to the same synonym set represented by WordNet, or is the paraphrase of word in the reference. For our word label tagging task, we opted TERp-A, another version of TERp, in which each above-mentioned edit cost has been tuned to maximize the correlation with human judgment of Adequacy at the segment level (from the NIST Metrics MATR 2008 Challenge development data). Figure 3 illustrates the labels generated by TERp-A for one hypothesis and reference (post-edited sentence) pair.

Reference	The	consequence	of	the	fundamentalist	movement		also	has	its importance	.
		S			S	Y	I		D	P	
Hyp After Shift	The	result	of	the	hard-line	trend	is	also		important	.

FIGURE 3 – Example of training label task using TERp-A.

Each word or phrase in the hypothesis is aligned to a word or phrase in the reference with a type of edit: “I” for insertions, “S” for substitutions, “T” for stem matches, “Y” for synonym matches, “P” for phrasal substitutions, and “D” for deletions. We do not consider words marked with “D” since they appear only in the reference. The lack of a symbol indicates an exact match (we replace it with “E” thereafter). Since our objective in this work is to train a binary classifier, we re-categorize the obtained 6-label set into binary set: The E, T and Y are regrouped into Good category, whereas the S, P and I belong to the Bad category. Finally, we observed that out of total words (in both of training and test sets) are 85% labeled “G”, 15% labeled “B”.

5.4 Classifier Selection

Among the various CRF toolkits, we selected WAPITI to train our CRF model as well as to tag the binary label for each word in the test set. WAPITI – developed by LIMSI-

[1] http://www.umiacs.umd.edu/~snover/terp/doc v1.html

CNRS - is based on maxent, maximum entropy Markov and linear-chain CRF models. It is well suited for huge feature sets up to several billions and allows us to gain significantly in training time.

The training phase was conducted on our 10000 sentence set. In all experiments with different feature sets, we applied uniquely the Stochastic Gradient Descent (SGD) algorithm for L1-regularized model, which works by computing the gradient only on a single sequence at a time and making a small step in this direction, therefore it can quickly reach an acceptable solution for the model. In the train command, we set values for maximum number of iterations done by the algorithm (-maxiter), stop window size (--stopwin) and stop epsilon (--stopeps) to 200, 6, and 0.00005 respectively. We compared our binary classifier performance not only with the other ones, but also with two naive baselines that were previously created. In baseline 1, we labeled all words in the MT hypothesis as good translations. In baseline 2, we assigned them randomly into G or B with respect to the percentage between two labels like in the corpus (85% G, 15% B).

6 Experiments and Results

6.1 Evaluation Metrics

We evaluated the performance of our classifiers by using very common evaluation metrics: Precision, Recall and F-score. Suppose that we would like to calculate these values for label “B”. Let X be the number of words whose true label is B and have been tagged with this label by the classifier, Y is the total number of words classified as B, and Z is the total number of words which true label is B. Thanks to these concepts, Precision, Recall and F-score can be defined as follow:

$$\Pr = \frac{X}{Y} \quad Rc = \frac{X}{Z} \quad F = \frac{2 \times \Pr \times Rc}{\Pr + Rc} \tag{8}$$

These calculations can be applied in the same way for label “G”. It is straightforward to recognize that the higher precision is, the more precise our classification result will be. Meanwhile, the recall reflects our classifier's capability to retrieve the accurate label for words. F-score is the “harmonic balance” between the two.

6.2 Results and Analysis

We perform our preliminary experiment by training a unique classifier with the combination of all proposed features (21 features). The training algorithm and related parameters were discussed in Section 5.4. The values of precision and recall for “Good” and “Bad” label are tracked and their fluctuations corresponding to thresholds (from 0.3 to 1.0, step 0.025) are represented in Figure 3. Results indicate that in case of Bad label, recall increases nearly monotonously when threshold is enlarged incrementally (except the huge fluctuation from 0.58 to 1 when threshold reaches 1), whereas precision falls from 0.42 to 0.18. With Good label, the variation occurs in the opposite direction: recall drops almost regularly from 0.92 to 0.78, then falls down to 0 in the final iteration, meanwhile precision goes up marginally from 0.848 to 0.881.

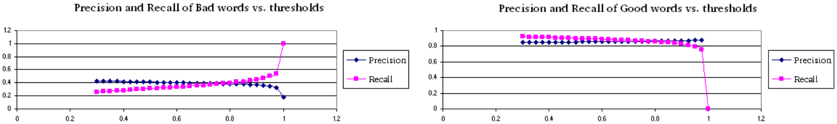


FIGURE 3 – Precision and Recall of labels vs. thresholds.

The curves representing the relationship between precision and recall of each class can be observed in Figure 4.

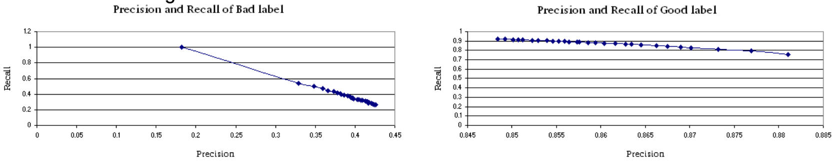


FIGURE 4 – Relationship between Precision and Recall of each label.

Table 3 reports the average values of Precision, Recall and F-score of these labels in the “all-features” system and the baseline systems. Results observed suggest that: (1) Good label is much better predicted than Bad label, (2) The combination of features helped to improve significantly the classifier’s capacity to detect the translation errors (which the improvement of 28.55% in terms of F score for B label comparing with baseline 2).

System	Label	Pr(%)	Rc(%)	F(%)
All features	Good	86.01	87.47	86.66
	Bad	38.81	38.05	36.78
Baseline 1	Good	100.00	94.48	97.14
	Bad	0.00	-	-
Baseline 2	Good	85.23	94.47	89.61
	Bad	15.08	5.66	8.23

TABLE 3 – Average Precision, Recall and F-score for labels.

Compare to the result of (Bach et al., 2011), our F-score for G label is 11.16% better, however they outperform us in F-score for B label (27.02% higher). According to our analysis, this might be originated from the following reasons: (1) our training and testing corpus are much smaller than theirs (10.8K vs. 75K) and differ about language pairs, (2) in our corpus, the percentage of G words overwhelms B words (85% vs. 15%) and (3) the best combination of features has not been investigated yet in this paper. All of these issues will be further considered in our future work.

7 Conclusions and Perspectives

We presented an approach to confidence estimation at word level for machine translation which explores various kinds of features, including those from the MT system together with those related to lexical and syntactic function of word in a sentence. A CRF based model has been investigated to train these above features and form our binary classifier. Experimental results show that precision and recall obtained in Good label are very promising, and can be acceptable in Bad label. More meaningful scores are hopefully still ahead with a deeper investigation in each separated feature as well as their various combinations. The comparison with baselines system demonstrates enormous contributions of features towards the perfectibility of the classifier. We employed TERp-A toolkit to generate word labels which is better correlated to human judgment, then regrouped them to a binary set.

In future, this work can be extended in the following ways. Firstly, we plan to conduct the “feature selection” strategy to sort our feature set in the ascending order of their usefulness. From this result we will have better understanding about each feature and its combination with others, as well as eliminate those who are not interesting. Besides of this, we will investigate another type of feature named semantic feature based on some other knowledge resources like WordNet which hopefully can help to improve our state-of-the-art classifier’s performance in terms of F-score, especially for Bad Label set. Another task will also be focused on is to find the most optimized methodology to conclude the confidence of whole sentence relied partially on the word-level confidence obtained from this current work.

References

ALBERTO SANCHIS, ALFONS JUAN, and ENRIQUE VIDAL (2007). Estimation of confidence measures for machine translation. In *Proceedings of the MT Summit XI*, Copenhagen, Denmark.

BONNIE DORR MATTHEW SNOVER, NITIN MADNANI and RICHARD SCHWARTZ (2008). TERp system description. In *MetricsMATR workshop at AMTA*.

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.

JESUS GIMENEZ and LLUIS MARQUEZ (2010b). Linguistic Features for Automatic MT Evaluation. *To Appear in 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.

J. LAFFERTY, A. MCCALLUM, and F. PEREIRA (2001). Conditional random fields: Probabilistic models for segmenting et labeling sequence data. In *Proc. ICML*.

LUCIA SPECIA, MARCO TURCHI, NICOLA CANCEDDA, MARC DYMETMAN, and NELLO CRISTIANINI (2009). Estimating the Sentence-Level Quality of Machine

Translation Systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37, Barcelona.

LUCIA SPECIA, ZHUORAN WANG, MARCO TURCHI, JOHN SHAWETAYLOR, and CRAIG SAUNDERS (2009). Improving the confidence of machine translation quality estimates. *In Proceedings of the MT Summit XII*, Ottawa, Canada.

NGUYEN BACH, FEI HUANG and YASER AL-ONAIZAN (2011). Goodness: A method for measuring Machine Translation Confidence. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 211–219, Portland, Oregon, June.

NICOLA UEFFING and HERMANN NEY (2007). Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.

NICOLA UEFFING and HERMANN NEY (2005). Word-level confidence estimation for machine translation using Phrased-based translation models. *Proceedings HLT/EMNLP*, pages 763–770, Vancouver.

P. KOEHN, H. HOANG, A. BIRCH, C. CALLISON-BURCH, M. FEDERICO, N. BERTOLDI, B. COWAN, W. SHEN, C. MORAN, R. ZENS, et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL'07*, pages 177–180, Prague, Czech Republic, June.

PETER F. BROWN, STEPHEN A. DELLA PIETRA, VINCENT J. DELLA PIETRA and ROBERT L. MERCER (1993a). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

POTET MARION, EMMANUELLE ESPERANÇA-RODIER, LAURENT BESACIER and HERVE BLANCHON (2012). Collection of a Large Database of French-English SMT Output Corrections. In *Proceedings of the eighth international conference on Language Resources and Evaluation (LREC).* Istanbul, Turkey, May.

POTET MARION, LAURENT BESACIER and HERVÉ BLANCHON (2010). The LIG machine translation system for WMT 2010. In *Proceedings of the joint fifth Workshop on Statistical Machine Translation and Metrics MATR (WMT2010)*, ACL Workshop. Uppsala, Sweden. 11-17 July.

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.

S. GANDRABUR and G. FOSTER (2003). Confidence estimation for text prediction. *In Proceedings of CoNLL*, Edmonton, May.

STOLCKE, A. (2002), SRILM - an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*, Denver, USA, pp. 901-904.

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.

THOMAS LAVERGNE, OLIVIER CAPPE and FRANÇOIS YVON (2010). Practical very large scale CRFs. In *Proceedings ACL*, pages 504–513.