Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com

Cascaded Re-ranking Modelling of translation hypotheses using extreme learning machines

Chi Man Vong[a,*], Yan Liu[a], Jiuwen Cao[b], Chun Yin[c]

[a] *Department of Computer and Information Science, University of Macau, Macau*
[b] *Institute of Information and Control, Hangzhou Dianzi University, Zhejiang 310018, China*
[c] *School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

ARTICLE INFO

Article history:
Received 1 September 2016
Received in revised form 5 February 2017
Accepted 1 May 2017
Available online xxx

Keywords:
Cascaded Re-ranking Modelling
Extreme learning machine
Statistical machine translation

ABSTRACT

In statistical machine translation (SMT), re-ranking of huge amount of randomly generated translation hypotheses is one of the essential components in determining the quality of translation result. In this work, a novel re-ranking modelling framework called *Cascaded Re-ranking Modelling* (CRM) is proposed by cascading a classification model and a regression model. The proposed CRM effectively and efficiently selects the good but rare hypotheses in order to alleviate simultaneously the issues of translation quality and computational cost. CRM can be partnered with any classifier such as *support vector machines* (SVM) and *extreme learning machine* (ELM). Compared to other state-of-the-art methods, experimental results show that CRM partnered with ELM (CRM-ELM) can raise at most 11.6% of translation quality over the popular benchmark Chinese–English corpus (IWSLT 2014) and French–English parallel corpus (WMT 2015) with extremely fast training time for huge corpus.

1. Introduction

In SMT, given a trained language model and a translation model, source sentences are firstly decoded into an *N*-best list consisting of huge amount of translation hypotheses along with their features. By comparing the hypotheses with their corresponding target sentences, the translation qualities of these hypotheses are evaluated and expressed in real values called BLEU (**Remark**: BLEU stands for Bilingual Evaluation Understudy [1], which is a measure for the quality of translation hypothesis from one natural language to another). Given the features in *N*-best list and some additional features as training inputs and the translation qualities as target outputs, a re-ranking model is traditionally constructed by regression as shown in Fig. 1. The job of re-ranking model is two-folded: (i) evaluate the qualities of future translation hypotheses; (ii) select the hypothesis of best quality as final translation result. Obviously, re-ranking model is an essential component in SMT. Therefore, it is a popular research topic to construct an effective and efficient re-ranking model (called re-ranking modelling). In execution phase, a source sentence is firstly decoded into a list of hypotheses along with their corresponding features. Then the constructed re-ranking model is applied on these features to predict the translation qualities, i.e., BLEU values, of the list of hypotheses. According to the predicted BLEU values, the best translation result is selected from the list of hypotheses. The execution phase is depicted in Fig. 2.

In the literature, re-ranking modelling can be done through different ways. Linguistics knowledge approach [2–6] is effective but human experts in linguistics are highly involved. Discriminative training [7,8] uses machine learning method rather than linguistics knowledge in re-ranking modelling which focuses on finding the relationships between hypothesis features and the translation quality such as BLEU. However, re-ranking tasks are very complex and highly nonlinear so that simple linear model [9,10] or log-linear model [11–13] are incapable to identify such relationship. To alleviate this issue, one approach is using boosting method [8,14,15,5] but not very effective. Another approach is using nonlinear machine learning model to construct re-ranking model such as *Markov random fields* [8], *nonlinear regression* [7,16,17], *support vector machine* (SVM) [18,19], and *multilayer perceptron* [20,21].

In whatever way, millions (or even more) of translation hypotheses can be easily generated from just thousands of source sentences, leading to heavy computation burden for re-ranking modelling. Recently, a kind of *single layer feedforward neural network* (SLFN) called *extreme learning machine* (ELM) [22–29] was proposed, whose input weights and bias are randomly generated and output weights can be calculated without iteration. The ELM model training time is extremely fast even for huge amount of training data. From its properties, ELM is very suitable for re-ranking modelling in discriminative training. However, ELM is a universal tool that is not specifically designed for re-ranking modelling. An effective framework utilizing ELM for re-ranking modelling is hence desired where high accuracy and low computational complexity are expected.

In fact, among these millions of translation hypotheses, only a few are of high quality while the remaining are poor or unnecessary for

* Corresponding author.
Email addresses: cmvong@umac.mo (C.M. Vong); senliuy@gmail.com (Y. Liu); jwcao@hdu.edu.cn (J. Cao); yinchun.86416@163.com (C. Yin)

http://dx.doi.org/10.1016/j.asoc.2017.05.002
1568-4946/

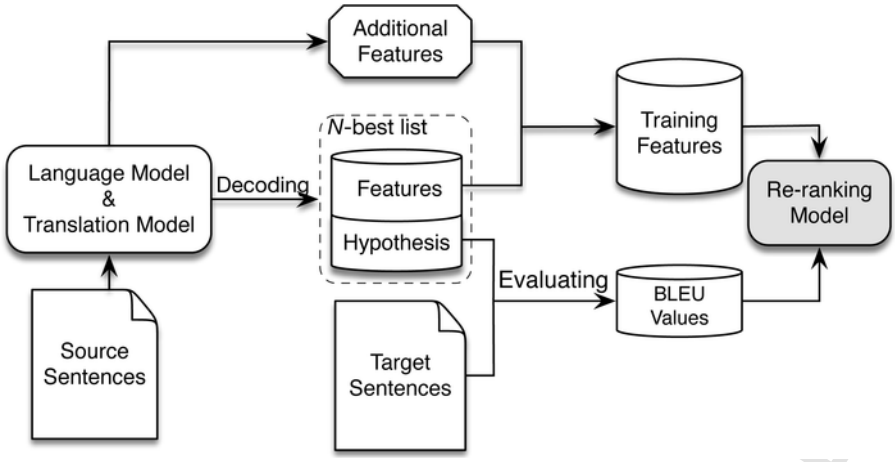

Fig. 1. Training phase of re-ranking model in SMT.

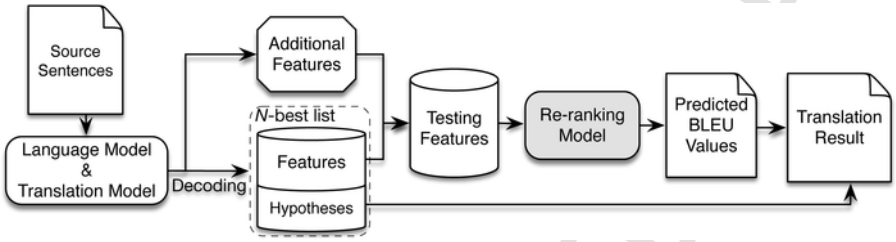

Fig. 2. Execution phase of re-ranking model in SMT.

re-ranking modelling. Inspired from this idea, we propose a novel framework which cascades classification and regression that (i) firstly filters out the large amount of poor or unnecessary hypotheses through classification; (ii) subsequently constructs the re-ranking model through regression based on the very few amount of high-quality hypotheses. This proposed framework is named as *Cascaded Re-ranking Modelling* (CRM), which improves the translation quality while simultaneously easing the computation burden for re-ranking modelling.

Basically, there are two versions of ELM: *plain* and *kernel* [30]. Plain ELM is very efficient for Big data while kernel ELM achieves higher stability and generalization but limited to smaller amount of training data due to memory issue. In this work, both versions are employed in CRM for classification and regression respectively. By utilizing ELM in CRM, it is called CRM-ELM.

In short, CRM-ELM has the following contributions for re-ranking modelling:

- Significant reduction of training time;
- Improvement of translation quality;
- Scalable to huge amount of translation hypotheses easily.

The rest of this paper is organized as follows. In Section 2, ELM classification and regression are briefly reviewed. In Section 3, the steps of CRM-ELM including data pre-processing, CRM-ELM training, and execution step are detailed. Experimental setup and results are provided in Section 4. Finally, conclusion is drawn in Section 5.

2. Review of extreme learning machine and Bilingual Evaluation Understudy

ELM was proposed as a kind of SLFN. The greatest advantage of ELM is its training procedure without iteration so that huge amount of training data can be handled efficiently. This advantage enables the extremely fast model training even for very large dataset, e.g., the *N*-best list in re-ranking task. In CRM-ELM, plain ELM classification and kernel ELM regression are employed for filtering and evaluation respectively, whose procedures are briefly reviewed in the following.

2.1. Plain ELM classification

Given a training dataset $\mathbf{D} = \{\boldsymbol{f}_i, t_i\}_{i=1}^{n}$ of n distinct hypotheses, $\boldsymbol{f}_i \in R^d$ are the d-dimensional input features and $t_i \in [-1, 1]$ (single output node) or $t_i \in [1, 2, \ldots, c]$ (c output nodes). An ELM classification model with m hidden nodes can be constructed as follows:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\beta_j g(\boldsymbol{w}_j \boldsymbol{f}_i + b_j) \tag{1}$$

where $\boldsymbol{w}_j$ and b_j are randomly generated input weights and bias, $\boldsymbol{\beta}_j$ are the output weights and $g(\cdot)$ is an activation function.

If the network with activation function $g(\cdot)$ can perfectly approximate the n hypotheses, the ELM model can be constructed as follows:

$$h(\boldsymbol{f}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\beta_j g(\boldsymbol{w}_j \boldsymbol{f}_i + b_j) \tag{2}$$

• Single output node (binary classification):

$$t_i = \text{sign}(h(\boldsymbol{f})) \tag{3}$$

• c output nodes (c-classes classification):

$$t_k = \arg \max_{k=\{1,\ldots,c\}} (h_k(f)) \tag{4}$$

$h_k(x)$ denotes the kth output node. And (2) can be written in matrix form:

$$\mathbf{G}\beta = \boldsymbol{h} \tag{5}$$

where

$$\mathbf{G} = \begin{bmatrix} g(\boldsymbol{w}_1 \cdot \boldsymbol{f}_1 + b_1) & \cdots & g(\boldsymbol{w}_m \cdot \boldsymbol{f}_1 + b_m) \\ \vdots & \cdots & \vdots \\ g(\boldsymbol{w}_1 \cdot \boldsymbol{f}_n + b_1) & \cdots & g(\boldsymbol{w}_m \cdot \boldsymbol{f}_n + b_m) \end{bmatrix} \tag{6}$$

$\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)^T$ and $\boldsymbol{t} = (t_1, t_2, \ldots, t_n)^T$. In ELM approach, the input weights $\boldsymbol{w}_i$ and hidden layers bias b_i are randomly generated at the beginning of training procedure. Hence, hidden layer output matrix $\mathbf{G}$ remains unchanged during training. For fixed $\boldsymbol{w}_i$ and b_i, training of ELM means finding a least-square solution $\widehat{\beta}$ to minimize the error function:

$$E(\boldsymbol{w}_i) = \sum_{j=1}^{n} I\left(\left\| \sum_{i=1}^{m} \beta_i g(\boldsymbol{w}_i \boldsymbol{f}_j + b_i) - t_j \right\| \right) \tag{7}$$

where

$$I(x) = \begin{cases} 0 & \text{if x} = 0 \\ 1 & \text{if x} \neq 0 \end{cases} \tag{8}$$

By using Moore–Penrose generalization inverse $\mathbf{G}^{\dagger} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T$ of matrix $\mathbf{G}$, the output weights $\boldsymbol{\beta}$ can be calculated from $\boldsymbol{w}_i$ and b_i without iteration. The solution for output weights is denoted as $\widehat{\beta} = \mathbf{G}^{\dagger}\boldsymbol{t}$. In summary, the following three steps are performed in order to obtain an ELM classification model:

i) Randomly generate input weights $\boldsymbol{w}_i$ and bias b_i;
ii) Calculate the hidden layer output matrix $\mathbf{G}$ in (6) and obtain $\mathbf{G}^{\dagger}$;
iii) Calculate the output weight $\boldsymbol{\beta}$ by

$$\beta = \boldsymbol{G}^{\dagger}\mathbf{t} \tag{9}$$

2.2. Kernel ELM regression

The output of an ELM regression model is a single value rather than multiple outputs in ELM classification. According to the theory of ELM [22,23], almost all non-linear piecewise continuous functions can be expressed by SLFN using kernel function κ such as Gaussian kernel with user-defined paramter σ,

$$\kappa(\boldsymbol{f}, \boldsymbol{f}') = \exp\left(\frac{\|\boldsymbol{f} - \boldsymbol{f}'\|_2^2}{2\sigma^2} \right) \tag{10}$$

By κ, we can define a kernel matrix Ω for ELM as follows:

$$\Omega_{i,j} = \kappa(\boldsymbol{f}_i, \boldsymbol{f}_j) \tag{11}$$

which can be further written in a matrix form

$$\Omega = \mathbf{G}\mathbf{G}^T \tag{12}$$

where

$$\mathbf{G} = \begin{bmatrix} \kappa(\boldsymbol{f}, \boldsymbol{f}_1) \\ \cdots \\ \kappa(\boldsymbol{f}, \boldsymbol{f}_n) \end{bmatrix} \tag{13}$$

With hidden layer output matrix $\mathbf{G}$ and the matrix of target output values $\mathbf{B}$. The ELM regression model is expressed as

$$\begin{aligned} \mathbf{M}(\boldsymbol{f}) &= \mathbf{G}^T \left(\frac{\mathbf{I}}{C} + \mathbf{G}\mathbf{G}^T \right)^{-1} \mathbf{B} \\ &= \begin{bmatrix} \kappa(\boldsymbol{f}, \boldsymbol{f}_1) \\ \cdots \\ \kappa(\boldsymbol{f}, \boldsymbol{f}_n) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \Omega \right)^{-1} \mathbf{B} \end{aligned} \tag{14}$$

where C is an user-defined regularization factor and $\mathbf{I}$ is an identity matrix.

2.3. Bilingual Evaluation Understudy (BLEU)

The evaluation of a hypothesis means assigning a real value according to how well the hypothesis represents its original meaning. Traditional method employs human translators. Although the best result is achieved using human translators, the evaluation is not automated and becomes impractical for huge amount of hypotheses.

Currently, the most promising automated evaluation is *BiLingual Evaluation Understudy* (BLEU) [1], which is an evaluation metric considering both the words and their orders in hypothesis. BLEU uses hypothesis and its corresponding target sentence provided by human translators, and then generates a real value by calculating the n-gram precision and relative length between the hypothesis and its target sentence. N-gram precision means the number of n-grams in both the hypothesis and its target sentence. The precision can be individual which is the n-gram accuracy for a given value n or cumulative for values from 1 to n. Let α and γ be the lengths of the hypothesis and the target sentence, respectively. The *brevity penalty* (BP) is computed with:

$$\text{BP} = \begin{cases} 1 & \text{if } \alpha > \gamma \\ e^{\left(1 - \frac{\gamma}{\alpha}\right)} & \text{if } \alpha < \gamma \end{cases} \tag{15}$$

The BLEU values is calculated as:

$$\mathrm{BLEU} = \mathrm{BP} \cdot e^{\sum_{i=1}^{n} \lambda_i \log(\rho_i)} \tag{16}$$

where λ_i is the weight of different n-gram models with $\sum_{i=1}^{n} \lambda_i = 1$. In current experiment, $\lambda_i = \frac{1}{n}$.

3. Proposed method

3.1. Overview

Given a corpus of source and target sentences, the obtained training data $\mathbf{D} = \{\mathbf{H};\mathbf{F};\mathbf{B}\}$ is generated through a trained language model and a trained translation model respectively. Usually these two trained models are available in existing software package such as Moses [31]. $\mathbf{D}$ consists of the translation hypotheses $\mathbf{H}$, their corresponding features $\mathbf{F}$, and their corresponding hypotheses qualities $\mathbf{B}$ (i.e., BLEU values). The training procedure in CRM-ELM is depicted in Fig. 3 and listed as follows:

- Data pre-processing: Two sets of training data $\{\mathbf{F}_C, \mathbf{T}_C\}$ and $\{\mathbf{F}_R;\mathbf{B}_R\}$ are respectively extracted from $\mathbf{D}$, followed by two sub-procedures of feature normalization and label assignment (to be detailed in subsequent sections).
- Classification modelling: The classification model $\mathbf{M}_C$ for filtering out low quality hypotheses is trained using plain ELM over $\{\mathbf{F}_C, \mathbf{T}_C\}$ because of its extremely fast speed for large amount of data.
- Evaluation modelling: The regression model $\mathbf{M}_R$ for quality evaluation is trained using kernel ELM with $\{\mathbf{F}_R;\mathbf{B}_R\}$ because of its high generalization.

During training phase, the classification model $\mathbf{M}_C$ and regression model $\mathbf{M}_R$ are trained simultaneously.

While in execution or testing phase, $\mathbf{M}_C$ and $\mathbf{M}_R$ work sequentially. Given the normalized features of a hypothesis list decoded from a source sentence, $\mathbf{M}_C$ firstly filters out low quality hypotheses. Then only few amount of high quality hypotheses are input to $\mathbf{M}_R$ for evaluation, from which the best translation result is identified.

The following subsections detail the data pre-processing including feature normalization and label assignment, training, and execution procedures in CRM-ELM.

3.2. Data pre-processing

There are two main tasks in data pre-processing, namely, feature normalization and label assignment. In this stage, the training data for regression model $\mathbf{M}_R$ is also prepared.

3.2.1. Feature normalization

In SMT, the amount of hypotheses decoded from a source sentence is arbitrary because of the length of different sentences. In the experiment data $\mathbf{D}$, at most N translation hypotheses are decoded from a source sentence. For s source sentences, there are $m \le (s \times N)$ items in $\mathbf{D}$. Each item is constituted with the translation hypothesis h and its features $\boldsymbol{f}$ (both generated by the software Moses [31]), i.e., $\mathbf{D} = \{\mathbf{H};\mathbf{F};\mathbf{B}\} = \{h_i;\boldsymbol{f}_i;\boldsymbol{b}_i\}$, $i = 1, 2, \ldots, m$, where $\boldsymbol{f}_i = [f_{i1}, \ldots, f_{i\tau}]$ consists of τ different features.

Before training, all feature values $\boldsymbol{f}_i$ are simply normalized into [0, 1]:

$$f_{i,j}^{\mathrm{new}} = \frac{f_{i,j} - \min(\boldsymbol{f}_i)}{\max(\boldsymbol{f}_i) - \min(\boldsymbol{f}_i)} \tag{17}$$

where $j = 1, 2, \ldots, \tau$ and $max(\boldsymbol{f}_i)$ and $min(\boldsymbol{f}_i)$ are respectively the maximum and minimum of $\boldsymbol{f}_i$. $f_{i,j}^{\mathrm{new}}$ is called "scaled features". After normalization, the normalized features $\mathbf{F}_C = [\boldsymbol{f}_i^{\mathrm{new}}]$ is generated for classification model $\mathbf{M}_C$.

3.2.2. Label assignment

Although the training features $\mathbf{F}_C$ for translation hypotheses is generated, there are no corresponding class labels for $\mathbf{F}_C$. In order to train $\mathbf{M}_C$, every translation hypothesis needs to be assigned a label

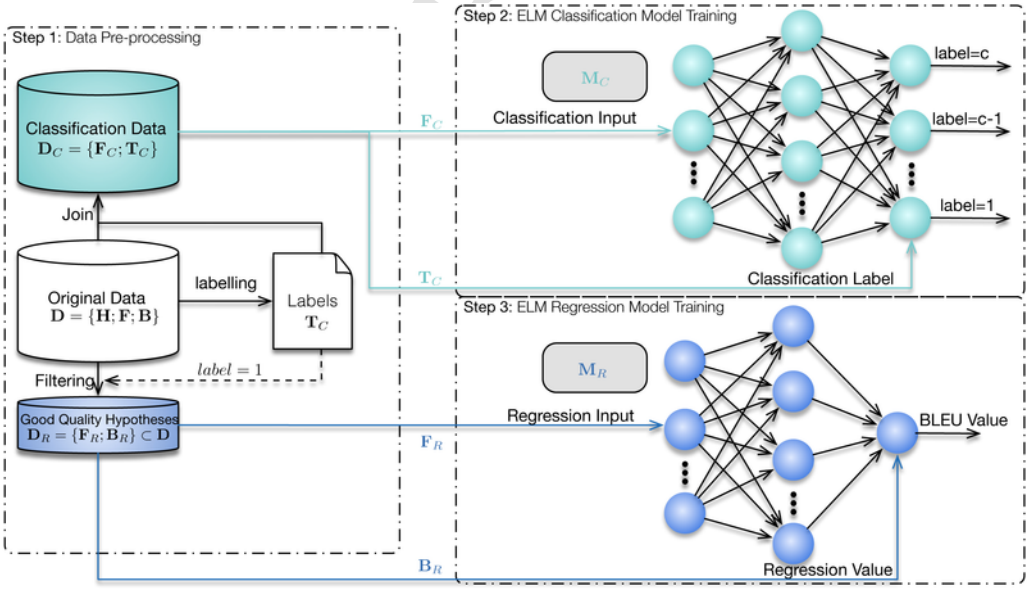

Fig. 3. Training procedure of CRM-ELM. In labelling step, $\mathbf{T}_C$ are automatically assigned according to $\mathbf{B}$. In join step, $\mathbf{F}_C$ is scaled from $\mathbf{F}$ and joined with $\mathbf{T}_C$. In filtering step, only the hypotheses of "$label = 1$" are selected into $\mathbf{D}_R$.

according to its corresponding translation quality $b \in \mathbf{B}$ (i.e., BLEU value). Hence, we need to determine the boundaries of BLEU values among classes, as illustrated in Fig. 4, for automatic label assignment. Suppose there are c classes or labels where c is a user defined parameter. It is equivalent to identify $c-1$ boundaries which can be expressed in a vector $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_{c-1}\}$.

For a source sentence having $m \leq N$ hypotheses, the m hypotheses are sorted according to their BLEU values b_p, $p = 1, 2, \ldots, m$, in descending order using Quicksort with time complexity of $O(m \cdot log(m))$. These m sorted BLEU values are then combined into an array $\mathbf{A}$,

$$\mathbf{A} = \text{Quicksort}(b_1, b_2, \ldots b_m, '\text{descending}') \tag{18}$$

Then θ_k is the $(\left\lceil \frac{m}{c} \right\rceil \cdot k)$th BLEU value in the array $\mathbf{A}$, where $k = 1, 2, \ldots c-1$. Subsequently, the label t_p for the pth hypothesis ($p = 1, 2, \ldots, m$) can be determined by

$$t_p = \begin{cases} 1 & \text{if } \theta_1 < \mathrm{b_p} \leq 1 \\ j & \text{if } \theta_j < \mathrm{b_p} \leq \theta_{j-1} \quad j = 2, 3, \ldots, c-1 \\ c & \text{if } 0 \leq \mathrm{b_p} \leq \theta_{c-1} \end{cases} \tag{19}$$

All labels generated in this step construct a vector $\mathbf{t}_C$, which constitutes the training data $\mathbf{D}_C = \{\mathbf{F}_C; \mathbf{t}_C\}$ for the classification model $\mathbf{M}_C$.

3.2.3. Training data for regression model $\boldsymbol{M}_R$

The regression model $\mathbf{M}_R$ is trained using only the best class of hypotheses. Simply speaking, its training data $\mathbf{D}_R = \{\mathbf{F}_R; \mathbf{B}_R\}$ are the features $\mathbf{F}_C$ with *label* = 1 (i.e., best quality):

$$\mathbf{F}_R = \{\boldsymbol{f}_i | t_i = 1\} \subset \mathbf{F}_C \quad i = 1, 2, \ldots, n; \tag{20}$$

$$\mathbf{B}_R = \{b_i | t_i = 1\} \subset \mathbf{B}_C \quad i = 1, 2, \ldots, n; \tag{21}$$

The pseudo code of data pre-processing is summarized in Algorithm 1.

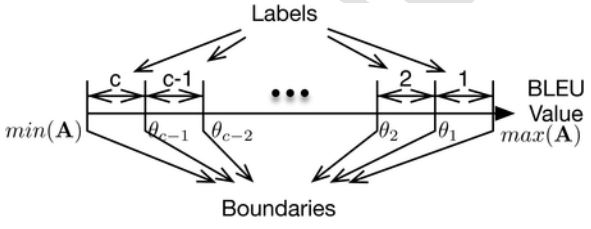

Fig. 4. The illustration of label assignment method.

Algorithm 1
Pseudo code of data pre-processing.

Input: Source sentences **S**, target sentences **T**.
output: Training features for classification model $\mathbf{F}_C$; Target labels for classification model $\mathbf{T}_C$; Training features for regression model $\mathbf{F}_R$; Target values for regression model $\mathbf{B}_R$.

1: $\{\mathbf{S}, \mathbf{T}\} \xrightarrow{\text{SRILM\&Moses}} \{\mathbf{D} = [\mathbf{H}; \mathbf{F}; \mathbf{B}]\}$; ◁ Generating experiment data with SRILM and Moses
2: $\{\mathbf{F}\} \xrightarrow{\text{Normalization}} \{\mathbf{F}_C\}$; ◁ Normalizing features **F** into scaled features $\mathbf{F}_C$
3: $\{\mathbf{B}\} \xrightarrow{\text{Labelling}} \{\mathbf{T}_C\}$; ◁ Labelling all hypotheses according to their translation qualities (BLEU values)
4: $\mathbf{F}_R = \{\boldsymbol{f}_i | t_i = 1\} \subset \mathbf{F}_C$ $i = 1, 2, \ldots, n$; ◁ Selecting the features of the best translation hypotheses
5: $\mathbf{B}_R = \{\boldsymbol{b}_i | t_i = 1\} \subset \mathbf{B}$ $i = 1, 2, \ldots, n$; ◁ Selecting the BLEU values of the best translation hypotheses
6: **return** $\{\mathbf{F}_C; \mathbf{T}_C\}$, $\{\mathbf{F}_R; \mathbf{B}_R\}$;

3.3. Cascaded Re-ranking Modelling using ELM

The training of CRM-ELM is carried out in two steps:

i) Classification modelling: a classifier $\mathbf{M}_C$ is trained using plain ELM over $\mathbf{D}_C = \{\mathbf{F}_C; \mathbf{T}_C\}$;
ii) Regression modelling: a regression model $\mathbf{M}_R$ is trained using kernel ELM over $\mathbf{D}_R = \{\mathbf{F}_R; \mathbf{B}_R\}$.

Note that both steps can be run in parallel after data pre-processing.

3.3.1. Training of $\boldsymbol{M}_C$

Given a scaled training dataset of m_C training instances $\mathbf{D}_C = \{\mathbf{F}_C; \mathbf{T}_C\} = \{\boldsymbol{f}_i^{\text{new}}, \boldsymbol{t}_i\}, i = 1, 2, \ldots, m_C$, where $\mathbf{F}_C$ and $\mathbf{T}_C$ are the scaled features in Section 3.2.1 and labels in Section 3.2.2, respectively. $\mathbf{M}_C$ can be trained in the following steps:

i) Select appropriately the activation function g_C and the number of hidden nodes L_C.
ii) Randomly generate an input weight matrix $\mathbf{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{L_C}\}$ and bias $\boldsymbol{b} = \{b_1, b_2, \ldots, b_{L_C}\}$.
iii) Calculate hidden layer output matrix $\mathbf{G}_C$:

$$\mathbf{G}_C = \begin{bmatrix} g_C(\boldsymbol{w}_1 \boldsymbol{f}_1 + b_1) & \cdots & g_C(\boldsymbol{w}_{L_C} \boldsymbol{f}_1 + b_{L_C}) \\ \vdots & \vdots & \vdots \\ g_C(\boldsymbol{w}_1 \boldsymbol{f}_{m_C} + b_1) & \cdots & g_C(\boldsymbol{w}_{L_C} \boldsymbol{f}_{m_C} + b_{L_C}) \end{bmatrix} \tag{22}$$

iv) Calculate the output weights $\boldsymbol{\beta}_C$:

$$\beta_C = \mathbf{G}_C^{\dagger} \mathbf{T}_C \tag{23}$$

where $\mathbf{T}_C = \{t_1, t_2, \ldots, t_{m_C}\}$, $\mathbf{G}_C^{\dagger}$ is the Moore–Penroes generated inverse of $\mathbf{G}_C$, and $\mathbf{G}_C^{\dagger} = (\mathbf{G}_C^T \mathbf{G}_C)^{-1} \mathbf{G}_C^T$.

Finally, output the classification model $\mathbf{M}_C = \{\mathbf{W}; \boldsymbol{h}; g_C; \boldsymbol{\beta}\}$.

3.3.2. Training of $\boldsymbol{M}_R$

Suppose m_R is the number of instances in training dataset $\mathbf{D}_R = \{\mathbf{F}_R; \mathbf{B}_R\}$. A regression model $\mathbf{M}_R$ over $\mathbf{D}_R$ can be generated in the following steps:

i) Selecting a kernel function κ_R such as Guassian kernel and the regularization coefficient C.
ii) Calculate the hidden layer output matrix $\mathbf{\Omega}$.

$$\Omega_{i,j} = \kappa_R(\boldsymbol{f}_i, \boldsymbol{f}_j) \tag{24}$$

or in matrix format

$$\Omega = \mathbf{G}_R \mathbf{G}_R^T \tag{25}$$

where

$$\mathbf{G}_R = \begin{bmatrix} \kappa(\boldsymbol{f}, \boldsymbol{f}_1) \\ \cdots \\ \kappa(\boldsymbol{f}, \boldsymbol{f}_{m_R}) \end{bmatrix} \tag{26}$$

iii) Calculating the output weight $\boldsymbol{\beta}_R$,

$$\beta_R = \begin{bmatrix} \kappa(\boldsymbol{f}, \boldsymbol{f}_1) \\ \cdots \\ \kappa(\boldsymbol{f}, \boldsymbol{f}_{m_R}) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \Omega \right)^{-1} \mathbf{B}_R \tag{27}$$

Finally, the regression model $\mathbf{M}_R = \{\kappa_R, C, \boldsymbol{\beta}_R\}$ is generated.

3.4. Execution or testing

Given a source sentence, Moses is employed to generate a list of translation hypotheses with their corresponding features for this sentence. Then, CRM-ELM is applied to find the best one from this list of hypotheses based on their corresponding features. In other words, the features (rather than the hypotheses) are the input to CRM-ELM. Given the data $\mathbf{D} = \{\mathbf{H}, \mathbf{F}\}$, ($\mathbf{F} = \{f_1, f_2, \ldots, f_m\}$) of m features (Note that $\mathbf{D}$ is generated by one source sentence only), the execution procedure is done in the following steps:

i) Scaling features: apply feature normalization in Section 3.2.1.
ii) Ranking hypotheses: $\mathbf{M}_C$ is applied to predict labels for all hypotheses, i.e., selecting the subset $\mathbf{D}'$ of $\mathbf{D}$ with best quality (label = 1).
iii) Evaluating hypotheses: The selected subset $\mathbf{D}'$ (with label = 1) are input to the regression model $\mathbf{M}_R$ to predict their BLEU values.
iv) Output translation result: The hypothesis with highest predicted BLEU value is returned as the final translation.

The execution flowchart of CRM-ELM is shown in Fig. 5.

4. Experiments

In this section, the experiment to validate the proposed method are described, together with the property of the corpora (data). Experimental setup and the process of model parameters tuning are detailed. The performance evaluation of CRM-ELM against other popular re-ranking modelling methods in terms of model training time and accuracy are shown.

4.1. Hardware and software

All the experiments were carried out under Mac OS X Yosemite 10.10.5 with an Intel Core i5 of 3.4 GHz and memory of 24 GB 1600 MHz DDR3. In addition, the following software packages were employed in our work.

- **GIZA++ v2**: an implementation [32] of IBM models and Hidden Markov Model (HMM) based alignment algorithm for computing the word alignments.
- **SRILM**: A 5-gram language model with Kneser-Ney smoothing was trained with SRILM [33] on monolingual target side of the parallel corpus for each language pair.
- **Moses**: The decoder [31] to generate the N-best list for each input source sentence from the test data. And Moses is also used to evaluate each hypothesis.

4.2. Corpora (data)

The experiments were conducted on two most recent benchmarking corpora, namely, IWSLT 2014 Chinese–English parallel corpus [34] and WMT 2015 French–English parallel corpus [35]. We used GIZA++ and SRILM for phrase alignment. Subsequently, Moses is applied to generate an N-best list of translation hypotheses (details of processing can be found in [32,33,31]). In our experiments, *holdout cross validation* was employed to randomly select the training data while the remaining data was for testing purpose. For Chinese–English translation of IWSLT 2014, $N = 100$ best hypotheses for every source sentence were generated for re-ranking modelling. For translation of French–English pair (WMT 2015), $N = 2000$ best hypotheses for every source sentence were considered in order to prepare a larger dataset. The statistics of datasets is presented in Table 1 and the features in the N-best list are shown in Table 2.

4.3. Settings

With the toolkits mentioned in Section 4.1 and corpora mentioned in Section 4.2, we generated a 100-best list for IWSLT 2014 and a larger 2000-best list for WMT 2015. With feature normalization discussed in Section 3.2.1, a matrix with scaled features was generated. Subsequently, the BLEU value of each hypothesis is evaluated. Ac-

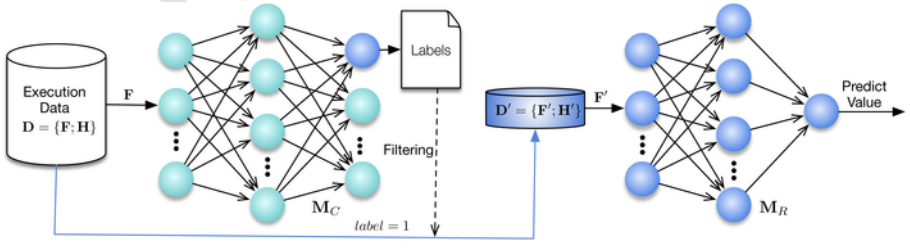

Fig. 5. Execution flowchart of CRM-ELM.

Table 1
Statistics of datasets.

Corpus	IWSLT 2014	WMT 2015
Language pair	English–Chinese	French–English
Test sentences	1093	3000
N-best hypotheses	100	2000
Collected hypotheses	109,040	3,213,906

Table 2
The features extracted for each hypothesis from *N*-best list.

Feature name	No. of values	Explanation
Reordering model	6	Models the influence of semantic relationships on the order of words within the scope of a phrase.
Distortion model	1	Measures the reordering of the input sentence. Higher reordering implies the translation is more expensive.
Language model	1	Measures the influence of this sentence. The statistical language model assigns a probability to a sequence of m words $P(\omega_1, \ldots, \omega_m)$ by means of a probability distribution.
Word penalty	1	Measures the word is too long or not.
Phrase penalty	1	The cost per produced phrase.
Translation model	4	Measure whether the translation result accords with the target language.
System score	1	The overall final score of the hypothesis produced by the model.

cording to its BLEU value, the label of each hypothesis is automatically assigned as described in Section 3.2.2. Finally, the dataset is randomly divided into training set and testing set. With the scaled features and the corresponding labels, a classification model $\mathbf{M}_C$ and a regression model $\mathbf{M}_R$ were constructed respectively. In CRM-ELM, the following parameters were empirically optimized.

- c, number of classes;
- g_C, activation function in model $\mathbf{M}_C$;
- L_C, number of hidden nodes in model $\mathbf{M}_C$;
- κ_R, kernel function in model $\mathbf{M}_R$;
- C, regularization coefficient.

A simple optimization method *Coordinate Descent*(CD) [36] was used to optimize the parameters. CD estimates the parameters by minimizing a selected parameter with all other parameters fixed. In our experiments, the scope of each parameter is shown in the following:

- c: binary or multi-classes (3, 5, 10, 15, 20);
- g_C: linear function, sigmoid function, sine function, hard limit function, triangular basis function, or radial basis function;
- L_C: 10–500, with step size equals to 10;
- κ_R: linear kernel, RBF kernel, polynomial kernel, or wavelet kernel;
- C: 0 to 1 with step size equals to 0.1.

For IWSLT 2014 and WMT 2015, the optimized activation function g_C is sigmoid and the optimized kernel function κ_R is RBF. The regularization coefficient C was optimally selected with 1. From experimental results, for IWSLT 2014 corpus, the class number $c = 10$ with 60 to 100 hidden nodes would be better (multi-class classification problem) while in WMT 2015, $c = 2$ (binary classification problem) and L_C would be 200 to 500.

4.4. Baseline

Baseline is simply a measure in BLEU that is considered as the lower bound on translation quality for a method. In SMT, MERT [37] (Minimum Error Rate Training) is always taken as the baseline because of its high accuracy. In our experiments, MERT was adopted as the baseline too. Details of MERT can be seen in [37].

4.5. Results

In order to fairly evaluate the effectiveness of CRM, both ELM and SVM were applied along with different sizes of training data. The experiments were run for 400 times, every time with a randomly re-selected training data. The experimental results are shown in Fig. 6 and Table 3. On average, CRM-ELM can raise up to 6.5% of BLEU value over baseline after parameters optimization. Another observation is that as training dataset gets larger, CRM-ELM tends to obtain a better result. When 1000 source sentences are used, the best re-ranking model achieves a relative improvement of 11.6% over baseline.

Another experiment was carried out on a bigger corpus, WMT 2015, containing totally 3009 source sentences. A 2000-best list was generated from WMT 2015, resulting totally more than 3 millions of hypotheses. In our experiments, the performance was found reaching the peak after the number of source sentences goes beyond 2000. Therefore, 2000 source sentences were randomly selected from WMT 2015 to generate the training dataset while the remaining source sentences become the test set. To verify the effectiveness of our proposed CRM, a state-of-the-art nonlinear discriminative training model called RankBoosting [10] was compared with CRM-ELM under WMT 2015 because both of these methods have similar properties. Under the same corpus, our proposed CRM-ELM obtains an

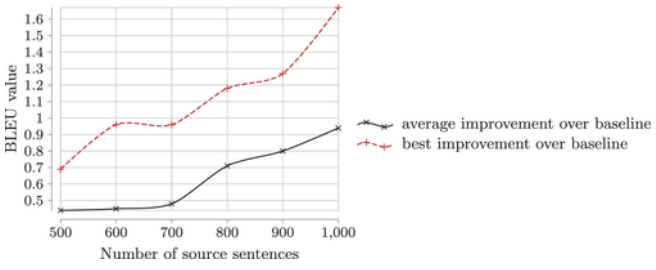

Fig. 6. Result of CRM-ELM for IWSLT 2014 over baseline.

Table 3
Re-ranking result (BLEU) of CRM-ELM for IWSLT 2014.

# Source sentences	Baseline	Experimental average	Standard deviation	Experimental best	Average improvement	Best improvement
500	14.69	15.13	±0.11	15.38	3.0%	4.7%
600	14.60	15.05	±0.15	15.56	3.1%	6.6%
700	14.84	15.32	±0.14	15.80	3.2%	6.5%
800	14.61	15.23	±0.18	15.79	4.9%	8.1%
900	14.55	15.35	±0.19	15.82	5.5%	8.7%
1000	14.40	15.34	±0.28	16.07	**6.5%**	**11.6%**

improvement of BLEU value of 0.59 (over baseline) while RankBoosting proposed in [10] has an improvement of just 0.346. In other words, our CRM-ELM is relatively 70% better under WMT 2015.

4.6. Comparison in translation quality

To illustrate the improvement on translation result of CRM for hypotheses re-ranking, the following four methods are compared with CRM-SVM and CRM-ELM:

- Plain ELM classification [22–25]: Plain ELM classification is compared to check the contribution of CRM framework. In plain ELM classification, the re-ranking model is evaluated under the percentage of misclassified labels.
- Kernel ELM regression: Regression for re-ranking modelling [7,16] is an effective approach on SMT. Therefore, kernel ELM regression [23] is employed for comparison.
- SVM voting [18]: SVM is an effective algorithm in machine learning. SVM voting provides a way for re-ranking modelling indirectly, in which a large margin classifier for sequential models is employed, which is similar to regression model.
- RankBoosting [10]: RankBoosting [38] is employed to construct a non-linear re-ranking model. We used 500 weak learners to construct the non-linear model just as what have been done in [10].

Six independent experiments were conducted according to different numbers of training source sentences (500, 600, 700, 800, 900, 1000) on IWSLT 2014. *Holdout cross validation* was employed here to select the training data and testing data.

For each of the six algorithms, 10 re-ranking models were trained after parameters optimization. Table 4 and Fig. 7 show the mean BLEU values of these algorithms. For SVM Voting, RankBoosting, CRM-SVM, and CRM-ELM, the best model from the 10 trained models are compared, whose results are shown in Fig. 8.

4.7. Comparison in training time

Another significant contribution of CRM-ELM is its extremely fast training time due to the advantageous learning mechanism of ELM. In our work, plain ELM classification, kernel ELM regression, SVM voting, RankBoosting, CRM-SVM, and CRM-ELM were compared in terms of training time.

The results are shown in Fig. 9, in which the training times of plain ELM classification, RankBoosting, and CRM-ELM are very short along increasing number of source sentences. However, the training time of kernel ELM regression, SVM voting and CRM-SVM increase exponentially. Although CRM-ELM takes longer time than plain ELM classification and RankBoosting, the training times of these three methods are just within seconds. Furthermore, our proposed CRM-ELM achieves the best translation quality. In SMT, translation quality is more important than training time and hence this slightly increased cost of training time is definitely worthy.

Afterwards, we tested a huge dataset of WMT 2015 with 3000 source sentence and 2000 best hypotheses, resulting 3 millions of hypotheses (some short sentences may produce less than 2000 hypotheses). SVM voting took more than one day to train the re-ranking model in super computer while CRM-ELM only takes less than one hour in personal computer.

5. Conclusions

Re-ranking is a significant component in SMT for translation quality and efficiency. A novel framework for re-ranking modelling called CRM is proposed. CRM cascades a classification model and a regression model for filtering out the large amount of low quality hypotheses. The few amount of high quality hypotheses are then employed to construct the re-ranking model through regression. Therefore, both translation quality and efficiency are ensured.

The significance of CRM is two-folded:

- Quality: CRM-ELM and CRM-SVM were both evaluated over two popular benchmark corpora, IWSLT 2014 (Chinese–English) and WMT 2015 (French–English). In our experiments, our proposed CRM-ELM can raise the translation quality (BLEU value) by 11.6% over baseline for IWSLT 2014, and 1.7% for WMT 2015, which is better than the state-of-the-art method [10] RankBoosting. For CRM-SVM, it has the similar translation quality to CRM-ELM except its training time is much longer.
- Efficiency: CRM can significantly reduce the training time over traditional method such as kernel ELM regression or SVM. The efficiency is obvious when CRM-ELM involves processing of millions of data. Therefore, ELM can be seen as the best partner of CRM.

Therefore, it can be concluded that the proposed CRM partnered with ELM can significantly improve both translation quality and efficiency for SMT which possesses a large amount of training data.

Acknowledgments

The work is financially supported by funding from University of Macau, project number MYRG2014-00083-FST, and from FDCT Macau, project number 050/2015/A, and was partial supported by the NNSF of China (61503104, 6153064).

Table 4
Comparison of different algorithms on BLEU value (IWSLT 2014 corpus).

# Training sentences	500	600	700	800	900	1000
Baseline	14.69	14.60	14.84	14.61	14.55	14.40
ELM classification	14.97±0.14	14.91±0.18	15.23±0.24	15.02±0.27	15.03±0.30	15.05±0.33
ELM regression	–	–	–	–	–	–
SVM voting	15.02 ± 0.08	15.01 ± 0.11	15.30 ± 0.14	15.11 ± 0.19	15.23 ± 0.23	15.12 ± 0.26
RankBoosting	15.05 ± 0.03	14.99 ± 0.02	15.28 ± 0.03	15.16 ± 0.12	15.28 ± 0.04	15.22 ± 0.14
CRM-SVM	15.12 ± 0.10	**15.06** ± 0.13	15.31 ± 0.19	15.20 ± 0.20	15.33 ± 0.23	**15.34** ± 0.27
CRM-ELM	**15.13** ± 0.11	15.05 ± 0.15	**15.32** ± 0.14	**15.23** ± 0.18	**15.35** ± 0.19	15.33 ± 0.28

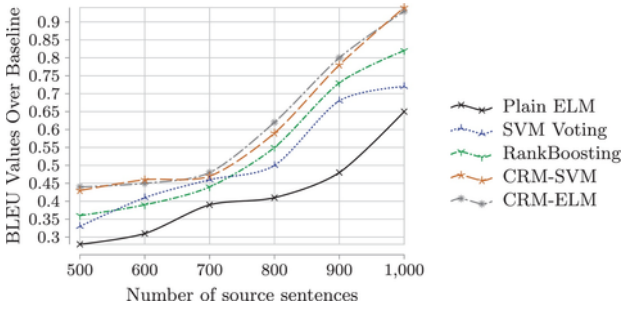

Fig. 7. Comparison of translation qualities over baseline under different methods on IWSLT 2014.

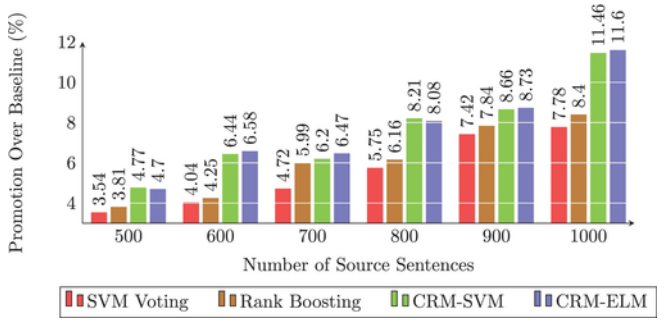

Fig. 8. Comparison of best BLEU values over state-of-the-art re-ranking models (IWSLT 2014).

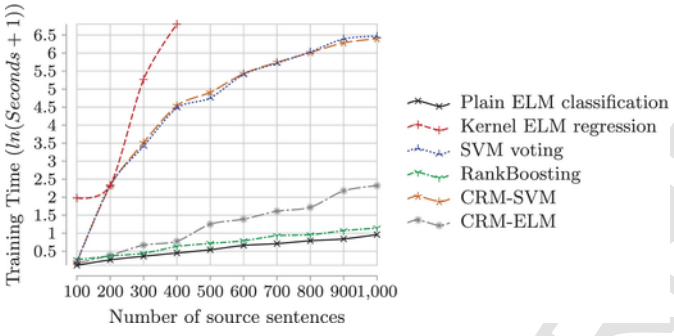

Fig. 9. Comparison in training time over ISWLT2014. Note that kernel ELM regression runs out of memory after 400 training source sentences.

References

[1] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2002, pp. 311–318.

[2] K. Kirchhoff, M. Yang, Improved language modeling for statistical machine translation, In: Proceedings of the ACL Workshop on Building and Using Parallel Text, Association for Computational Linguistics, 2005, pp. 125–128.

[3] Y. Zhang, A.S. Hildebrand, S. Vogel, Distributed language modeling for n-best list re-ranking, In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2006, pp. 216–223.

[4] A.L. Lagarda, F. Casacuberta, Applying boosting to statistical machine translation, In: Annual Meeting of European Association for Machine Translation (EAMT), 2008, pp. 88–96.

[5] N.-Q. Luong, L. Besacier, B. Lecouteux, Word confidence estimation for SMT n-best list re-ranking, In: Proceedings of the Workshop on Humans and Computer-assisted Translation (HaCaT) During EACL, 2014.

[6] Y. Wang, L. Wang, D.F. Wong, L.S. Chao, Effective Hypotheses Re-ranking Model in Statistical Machine Translation, Springer Berlin Heidelberg, Berlin/Heidelberg, 201424–32.

[7] L. Shen, A. Sarkar, F.J. Och, Discriminative reranking for machine translation, In: North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL), 2004, pp. 177–184.

[8] M. Collins, T. Koo, Discriminative reranking for natural language parsing, Comput. Linguist. 31 (1) (2005) 25–70.

[9] F.J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, et al., A smorgasbord of features for statistical machine translation, In: North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL), 2004, pp. 161–168.

[10] A. Sokolov, G. Wisniewski, F. Yvon, Non-linear n-best list reranking with few features, In: Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), San Diego, CA, 2012.

[11] F.J. Och, H. Ney, Discriminative training and maximum entropy models for statistical machine translation, In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2002, pp. 295–302.

[12] P.F. Brown, V.J.D. Pietra, S.A.D. Pietra, R.L. Mercer, The mathematics of statistical machine translation: parameter estimation, Comput. Linguist. 19 (2) (1993) 263–311.

[13] P. Koehn, F.J. Och, D. Marcu, Statistical phrase-based translation, In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, Association for Computational Linguistics, 2003, pp. 48–54.

[14] P. Nguyen, M. Mahajan, X. He, Training non-parametric features for statistical machine translation, In: Proceedings of the Second Workshop on Statistical Machine Translation, Association for Computational Linguistics, 2007, pp. 72–79.

[15] K. Duh, K. Kirchhoff, Beyond log-linear models: boosted minimum error rate training for n-best re-ranking, In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Paper, Association for Computational Linguistics, 2008, pp. 37–40.

[16] L. Shen, A. Joshi, Flexible margin selection for reranking with full pairwise samples, Nat. Lang. Process. – IJCNLP 2004 (2005) 446–455.

[17] D.S. Gonzalez-Gonzalez, R.J.P. Alejo, M. Cantú-Sifuentes, L.M. Torres-Trevi no, G.M. Méndez, A non-linear fuzzy regression for estimating reliability in a degradation process, Appl. Soft Comput. 16 (2014) 137–147.

[18] L. Shen, A.K. Joshi, An SVM based voting algorithm with application to parse reranking, In: Proceedings of the Seventh Conference on Natural Language Learning at Association for Computational Linguistics: Human Language Technologies(NAACL-HLT) 2003, vol. 4, Association for Computational Linguistics, 2003, pp. 9–16.

[19] L. Jedliński, J. Jonak, Early fault detection in gearboxes based on support vector machines and multilayer perceptron with a continuous wavelet transform, Appl. Soft Comput. 30 (2015) 636–641.

[20] L. Huang, Forest reranking: Discriminative parsing with non-local features, In: Association for Computational Linguistics (ACL), 2008, pp. 586–594.

[21] H. Sun, T. Zhao, Neural network-based reranking model for statistical machine translation, In: 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), IEEE, 2014, pp. 460–465.

[22] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, In: Proceedings. 2004 IEEE International Joint Conference on Neural Networks, 2004, vol. 2, IEEE, 2004, pp. 985–990.

[23] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 42 (2) (2012) 513–529.

[24] J. Tang, C. Deng, G.-B. Huang, Extreme learning machine for multilayer perceptron, IEEE Trans. Neural Netw. Learn. Syst. 27 (4) (2016) 809–821.

[25] G.-B. Huang, What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle, Cogn. Comput. 7 (3) (2015) 263–278.

[26] J. Cao, W. Huang, T. Zhao, J. Wang, R. Wang, An enhance excavation equipments classification algorithm based on acoustic spectrum dynamic feature, Multidimens. Syst. Signal Process..

[27] A.S. Alencar, A.R.R. Neto, J.P.P. Gomes, A new pruning method for extreme learning machines via genetic algorithms, Appl. Soft Comput. 44 (2016) 101–107.

[28] J. Cao, W. Wang, J. Wang, R. Wang, Excavation equipment recognition based on novel acoustic statistical features, IEEE Trans. Cybern.

[29] J. Cao, K. Zhang, M. Luo, C. Yin, X. Lai, Extreme learning machine and adaptive sparse representation for image classification, Neural Netw. 81 (2016) 91–102.

[30] G.-B. Huang, An insight into extreme learning machines: random neurons, random features and kernels, Cogn. Comput. 6 (3) (2014) 376–390.

[31] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Moses Herbst, Open source toolkit for statistical machine translation, In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL'07, Association for Computational Linguistics, Stroudsburg, PA, USA, 2007, pp. 177–180. http://dl.acm.org/citation.cfm?id=1557769.1557821.

[32] F.J. Och, H. Ney, A systematic comparison of various statistical alignment models, Comput. Linguist. 29 (1) (2003) 19–51.

[33] A. Stolcke, et al., SRILM – an extensible language modeling toolkit, In: INTERSPEECH, 2002.

[34] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, M. Federico, Report on the 11th international workshop on spoken language translation evaluation campaign, international workshop on spoken language translation 2014, In: Proceedings of the Eleventh International Workshop on Spoken Language Translation (IWSLT), Lake Tahoe, CA, 2014, pp. 2–17.

[35] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, P. Pecina (Eds.), Proceedings of the Tenth Workshop on Statistical Machine Translation, Association for Computational Linguistics, Lisbon, Portugal, 2015. http://aclweb.org/anthology/W15-30.

[36] S.J. Wright, Coordinate descent algorithms, Math. Program. 151 (1) (2015) 3–34.

[37] F.J. Och, Minimum error rate training in statistical machine translation, In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, Association for Computational Linguistics, 2003, pp. 160–167.

[38] Y. Freund, R. Iyer, R.E. Schapire, Y. Singer, An efficient boosting algorithm for combining preferences, J. Mach. Learn. Res. 4 (Nov) (2003) 933–969.