CLASSIFICATION OF MACHINE TRANSLATION OUTPUTS USING NB CLASSIFIER AND SVM FOR POST-EDITING

Kuldeep Kumar Yogi[1] , Chandra Kumar Jha[2] and Shivangi Dixit[3]

[1,2,3] Department of Computer Engineering, Banasthali University, Rajasthan, India

ABSTRACT

Machine translation outputs are not correct enough to be used as it is, except for the very simplest translations. They only give the general meaning of a sentence not the exact translation. As Machine Translation (MT) is gaining a position in the whole world, there is a need for estimating the quality of machine translation outputs. Many prominent MT-Researchers are trying to make the MT-System, that produces very good and accurate translations and that also covers maximum language pairs. If good translations out of all translations can be categorized then the time and cost can be saved to a great extent. Now, Good quality translations will be sent for post-editing and rest will be sent for pre-editing or retranslation. In this paper, Kneser Ney smoothing language model is used to calculate the probability of machine translated output. But a translation cannot be said good or bad. Based on its probability score there are many other parameters that effect its quality. The quality of machine translation is made easier to estimate for post-editing by using two different predefined famous algorithms for classification.

KEYWORDS

Machine Translation, Naïve Bayes Classifier, SVM, Kneser Ney Smoothing, MT-Quality Estimation, Post editing

1. INTRODUCTION

MT-Engine translate a inputted sentence according to its predefined inbuilt algorithm. Even the best translation systems make mistakes in translating the sentences, since till now, no such model exists which can accurately translate the sentence. Sometimes whole translations are fully meaningless. MT-Outputs generally contain grammatical errors, missing negations etc. (Sylvain Raybaud, 2009). Error analysis is very much time consuming. (Guillaume W.,Natalie K.,François Y. 2014). Human post-editing is required for only those sentences which have approximately 70% accurate translation outputs. (Kuldeep Y , 2015). In the translation process with machine translation (MT), post-editing cost much time and efforts on the part of human. There is no requirement of post-editing when MT-Outputs are of good quality.(Hirokazu Suzuki,2011).With the current submission, the problem of categorizing machine translation outputs is being addressed into two different categories automatically using classifiers. We have described how these two classifiers are classifying our MT-outputs in good and bad categories based on their different attributes values. This paper is consisted in these steps: The experiment and method is described in Section 3, including description about classifiers, language model and process. Section 4 describes the results whereas correlation with human judgement and feature conclusions are discussed in Section 5 and Section 6 respectively.

DOI : 10.5121/mlaij.2015.2403

2. RELATED WORKS

In the following paragraph, some of the related work and history in the area of Classification of MT-Outputs is briefly discussed using Classifiers. In 2014, Guillaume Wisniewski presented the corpus of MT errors that consists of post-edited translations with labels which are identifying the different types of errors of the MT system. In 2015, Shruti Tyagi has categorized the sentences into good and bad with the help of classifiers. She also concluded that Naïve Bayes is bit more better than support vector machines. Irina Rish(2001) demonstrated that Naïve Bayes is useful for two cases: for Completely Independent Features and Functionally Dependent Features. In 2009, Sylvain Raybaud presented various confidence scores for Statistical machine translation. Huang (2003) answered how do algorithms, such as decision trees and Naive Bayes are compared in terms of the better measure AUC(same as ROC -Receiver Operating Characteristics). In 2005, Thorsten Joachims introduced support vector machines for sentence categorization and concluded that SVMs are more suited for text categorization. In 2007, Simard, M., Goutte used phrase based MT-System for post-editing the translated output of a machine system and also tested the data on the job-bank data set.The difficulties faced by Eleftherios Avramidis in quality estimation of MT-outputs in 2012. Eleftherios worked on this problem in their research. A machine translated document can be more reliable after some modification but it should be automated to save human efforts. (Knight K., Ishwar C. ,1994).

3. EXPERIMENT SETUP

3.1. Corpus Selection & MT-Engines

A corpus collected of 1300 sentences of English Language related to tourism domain from various sites and magazines which are then translated into Hindi language by six Machine Translation systems.

Followings are the MT-Engines used for translation:

1. Babylon MT-Engine
2. Microsoft MT-Engine
3. EBMT MT-Engine
4. Anusharka MT-Engine
5. Moses MT-Engine
6. Joshua MT-Engine

3.2. Language Model

3.2.1 Kneser Ney Smoothing

Kneser-Ney smoothing is an extension of absolute discounting which was introduced by Kneser and Ney in 1995. We know that a lower order distribution is very important in the combined model only when few or more counts are present with the higher order distribution.

For example, a bigram model say San Francisco, We know that FRANCISCO occurs only after a single word SAN. Since C(FRANCISCO) is high, the unigram probability P(FRANCISCO) will be high and an algorithm such as absolute discounting will assign a relatively high probability to the word FRANCISSO. But this probability should not be high since the word FRANCISCO

follows a single word SAN. So the word FRANCISCO should receive a low unigram probability. In Kneser-Ney smoothing, we generalize this argument, not setting the unigram probability to be proportional t the number of occurrences of a word, but instead to the number of different words that it follow*s.*

Following formula is used to compute bigram probability:

$$P_{kn}(w_i|w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - \mathrm{d}, 0)}{c((w_{i-1}))} + \lambda(w_{i-1})Pcontinuation(w_i)$$

In this equation d is used for discount value it is reduced from the count of every n-gram, its value can be any between 0 and 1.

$\lambda(w_{i-1})$ is a Unigram Probability.

$Pcontinuation(w_i)$ is calculated for each word by how many words it completes in a corpus. Like in "Chandra Prakash" Chandra word may appear in corpus many times but how many times Chandra appears with Prakash, this is continuation probability. This count normalizes by dividing number of bigrams in corpus.

Source Sentence

Transit passengers who have a follow-on ticket within 72 hours of arrival as long as they are not going to leave the airport.

Target Sentence(Translated)

यात्रियों हैं जो एक टिकट के भीतर 72 घंटे की आगमन के रूप में लंबी रूप हैं वे नहीं छोड़ जाने के हवाई अड्डा है ।

Figure 1. English sentence and translated sentence in Hindi using EBMT

It is easy to understand KN smoothing using the example given in Figure1. The translated sentence's each word's bigram, unigram and continuation probability are computed here. Kneser have given a clever approach of continuation probability from which a more smoothed probability can be achieved.

3.3. Attributes

When some items or data are to be categorized into different classes or categories, some ideas based on which decision to put a particular item can be made in x, y or z etc. category. The idea can be a computation of various attribute values related to item. In the same way, here is taking MT-outputs for classification. 27 attribute values related to the translated sentence have computed. Based on these attribute's values, it has tried to classify MT-Engines-Output in good or bad categories.

Sentence's attribute-values are computed according to Table 1.

S NO.	Attribute/feature type
1.	Length of the source sentence
2.	Length of the target sentence
3.	Average source token length
4.	LM probability of source sentence
5.	LM probability of target sentence
6.	Average no. of occurrences of a target word within a target sentence
7.	Average of source word translations in a sentence
8.	Average of source word translations in a sentence (according to IBM table1, it is Prob(t|s)>0.01)
9.	Unigrams percentage in quartile1 of frequency in source language corpus
10.	unigrams percentage in quartile4 of frequency in source language corpus
11.	Bigrams percentage in quartile1 of frequency of source-words in source language corpus
12.	Bigrams percentage in quartile4 of frequency of source-words in source language corpus
13.	Trigrams percentage in quartile1 of frequency of source-words in source language corpus
14.	Trigrams percentage in quartile4 of frequency of source-words in source language corpus
15.	Unigrams percentage in the source sentence found in a corpus
16.	Count of punctuation-marks in source sentence
17.	Total punctuation-marks in a target sentence
18.	Number of mismatch between source and target punctuation-marks
19.	Number of content words in the source-sentence
20.	Number of content words in the target-sentence
21.	Content words percentage in the source-sentence
22.	Content words percentage in the target-sentence
23.	Number of non content words in the source-sentence
24.	Number of non content words in the target-sentence
25.	Non content words percentage in the source-sentence
26.	Non-content words percentage in the target-sentence
27.	LM probabilities of POS of target sentence

Table 1: Machine translation attributes.

3.4. Classifiers

3.4.1 Naive Bayes classification

Bayesian theorem is used in Naïve-Bayes(NB) classifier. It is suitable when input's dimensional is high. Naïve-Bayes invented a more simple classification method against already used complicated classification techniques. NB classifier is a probabilistic classifier that built from bayes algorithm. It is very simple and effective for text classification and used in spam detection, sexually explicit content detection, personal email sorting, and document categorization (Irina Rish, 2001) .It is less computationally intensive because it consumes less processor cycles, takes

less memory and small training data behind its similar techniques like Random Forests, Boosted Trees, Support Vector Machines Max Entropy etc.(Huang, 2003)
The NB classifier chooses the most likely classification V_{nb} mentioned in the attribute values a1, a2,.., an.

$$V_{nb} = \text{argmax}_{v_j \in V} P(v_j) \prod P(v_j \mid a_i)$$

Generally estimate $P(v_j \mid a_i)$ using m-estimates:

$$p(v_j \mid a_i) = \frac{P(a_i \mid v_j)P(v_j)}{p(a_i)}$$

$P(a_i \mid v_j)$ = probability of instance a_i being in class v_j ,
$P((v_j \mid a_i))$ = probability of generating instance a_i given class v_j ,
$P(a_i)$ = probability of occurrence of class a_i ,
$P(v_j)$ = probability of instance v_j occurring

The above mechanism of NB classifier to classify all MT-systems-outputs (1300*6 sentences) have been used into good and bad categories.

3.4.2 SVM

This mechanism was introduced in 1992. It was famous for recognizing the handwritten digit. Support Vector Machine is used to classify elements in 2 different classes like A and B. A boundary is used to categories the elements. This boundary is called Hyperplane. To estimate the boundaries, SVM uses several algorithms. It supports regression and classification. Support vector machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Mark-able performance can achieve using SVM in text categorization. There is not need of parameter tuning; it can set right parameter values automatically. (Thorsten Joachims, 2005)

3.5. Weka Toolkit

Weka is a collection of machine learning algorithms for data mining tasks. The Technique/algorithms can be set by writing own java programs or it can apply directly to dataset. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. Weka provides implementation of machine learning algorithms to classify the MT-Outputs. First Weka Toolkit needs to be installed and then all the required attributes needs to be fixed into it and finally, both the algorithms i.e. Naïve Bayes and SVM is applied into it to classify MT-Outputs in good and bad categories.

3.6. Process

The overall process starts from a client who will input a sentence for translation using web service. Client will get raw translation from MT-Engine. This translation is an input for the language model (LM). LM helps to compute the probability of the sentence. This probability score and some other attributes which are mentioned in Table 1 will pass in both the classifiers

i.e. Naïve Bayes(NB) Classifier and SVM. The classifier will classify the sentence in good or bad category according to given attribute's values. If the translation is good quality translation then it will be sent for post-editing otherwise it will be sent for pre-editing and retranslation.
This classification process will work according to following diagram:-

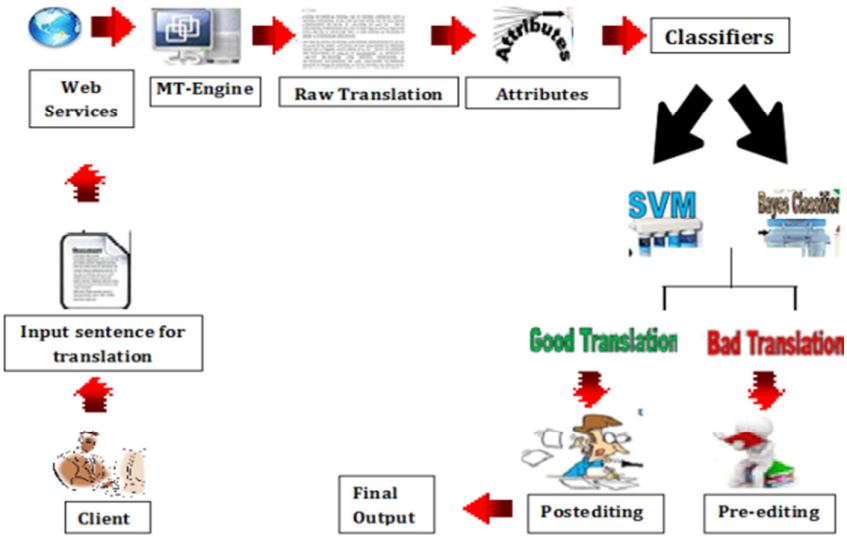

Figure 1. Overall system work flow

3.7. Result Analysis

i) The mechanism of NB classifier has been used to classify all MT-systems-outputs (1300*6 sentences). Sentence's attribute-values are computed according to Table 1. The results achieved are mentioned in Table 2 in Naïve Bayes Classifier using 27 different attributes values:-

Quality	Microsoft	Babylon	Anusharka	Moses	Joshua	EBMT
Excellent	68	56	59	47	62	65
Good	391	405	372	378	97	468
Average	781	787	770	844	626	738
Poor	60	52	99	31	515	29
Total	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**

Table 2. MT-Systems-output classification using Naïve Bayes's classifier

- Weka tool is the implementation of SVM in java language. It is freely available on the I Internet. This tool has been used to classify the data (sentences) into different categories like Excellent, Good, Average and Poor sentences. Following results are computed using SVM in Weka tool:-

Quality	Microsoft	Babylon	Anusharka	Moses	Joshua	EBMT
Excellent	22	26	28	5	16	21
Good	295	603	476	227	314	468
Average	969	627	697	967	767	764
Poor	14	44	99	101	203	47
Total	**1300**	**1300**	**1300**	**1300**	**1300**	**1300**

Table 3. MT-Systems-outputs classification using Support Vector Machine

ii) As per Table 2. EBMT and Babylon system are showing good translation outputs than other systems. Joshua MT-toolkit is showing very poor performance here and from Table 3., it can say that Babylon, Anusharka and EBMT are working very well but both MT-toolkits are not giving satisfactory results .So, it can be said that EBMT and Babylon MT-Toolkit is quite good in comparison to other toolkits.

iii) Excellent and good sentences have been calculated out of all 1300 sentences using both the classifiers i.e. Naïve Bayes Classifier & SVM. Table 4 represents total no. of good sentences. It can be understood more clearly using charts.(Figure 1.)

MT-Engines	Naïve Bayes Classifier	SVM
EBMT	533	489
Babylon	461	629
Microsoft	459	317
Anusharka	431	504
Moses	425	232
Josua	159	330

Table 4. Comparative estimation of no. of good translations by NB and SVM classifiers

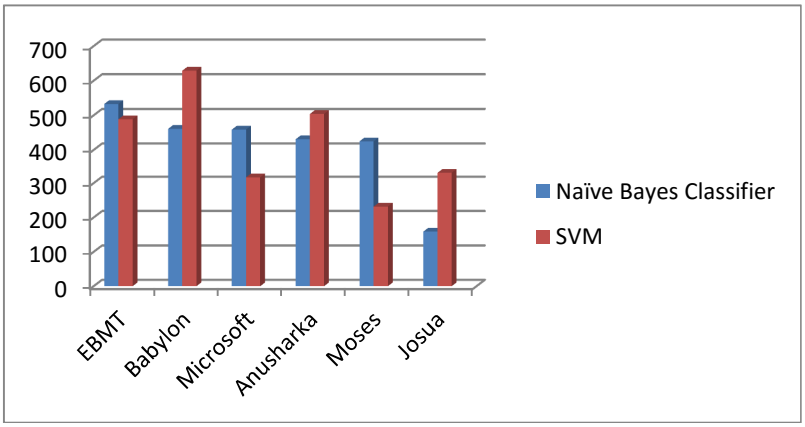

Figure2. Comparison of no. of good sentences through Classifiers using six different MT- Engines

Now, total 533 sentences can be sellable out of 1300 sentences approximately (good sentences) for post-editing, according to Naïve Bayes Classifier and around 489 sentences, according to Support Vector Machines and remaining sentences will be sent

for pre-editing to get good quality translation. In this way, the human post editing will be less expensive and quite fast. It will save the money and time to a great extent

4. CORRELATION WITH HUMAN

The result of NB classifier and SVM are correlated with human evaluation. There is a positive correlation with all Machine Translation systems. The highest correlation can be noticed with EBMT MT-Engine, it is 0.656024 and 0.65591 as mentioned in Table 5.

Classifier	Microsoft	Babylon	Anusarka	Moses	Joshua	EBMT
SVM	0.158707	0.515279	0.548851	0.472856	0.129962	0.656024
NB	0.158425	0.51498	0.548764	0.473045	0.13002	0.65591

Table 5: Correlation with human judgment

5. CONCLUSIONS

Human reference translations cannot be found, but still a good post editing candidate can be found. So, for this a machine learning measure needs to be employed. In this particular study, two classifiers were trained viz., an SVM based classifier and a Naïve Bayes classifier. 27 features were used for identifying the quality of MT outputs. In these, 18 feature were not required linguistic knowledge whereas 9 were used linguistic knowledge. 1500 sentences were used for training the classifiers using the outputs of 6 MT systems used in the study. One human evaluator's result was used to classify the outputs into two categories (good, poor). The computed values of both classifiers were correlated with human judgments that showed a good correlation with human evaluation. The correlations of two classifiers were also compared and it was found that among the two classifiers, naïve bayes produced better correlations with human judgments.
Linguistic resource was not found much for Indian languages in general and Hindi in particular. Some more linguistic resources like parsers, morphological analyzers, stemmers, POS taggers, etc. were need here so that some more semantic or semantic measures could be implemented. This could possibly give a posting measure which can provide results as good as human judgments.

REFERENCES

[1] Sylvain Raybaud, Caroline Lavecchia, David Langlois,and Kamel Sma¨ıli. 2009. *New Confidence Measures for Statistical Machine Translation*. Proceedings of the International Conference on Agents, pages 394–401.

[2] Guillaume Wisniewski, Natalie Kubler, François Yvon. 2014.*A Corpus of Machine Translation Errors Extracted from Translation Students Exercises*. In International Conference on Language Resources and Evaluation (LREC 2014), European Language Resources Association (ELRA), 2014.

[3] Kuldeep Kumar Yogi, Nishith Joshi,Chandra Kumar Jha. 2015. *Quality Estimation of MT-Engine Output Using Language Models for Post Editing and their Comparative Study*. Proceedings of Second International Conference INDIA 2015

[4] Suzuki, Hirokazu 2011. *Automatic Post-Editing based on SMT and its selective application by Sentence-Level Automatic Quality Evaluation*. Proceedings of the Machine Translation Summit XIII (2011), 156-163.

[5] Shruti Tyagi, Deepti Chopra, Iti Mathur, Nisheeth Joshi. (12 Jul 2015) *Classifier-Based Text Simplification for Improved Machine Translation*. In Proceedings of International Conference on Advances in Computer Engineering and Applications 2015.

[6] Jin Huang. 2003. *Comparing naive Bayes, decision trees, and SVM with AUC and accuracy.* Data Mining, 2003. ICDM 2003. Third IEEE International Conference on 19-22 Nov. 2003.
[7] Thorsten Joachims. 2005.*Categorization with Support Vector Machines: Learning with Many Relevant Features.*Volume 1398 of the series Lecture Notes in Computer Science pp 137-142
[8] Simard, M., Goutte, C., & Isabelle, P. (2007, April).*Statistical Phrase-based Post-editing*. Proceedings of NAACL HLT 2007, ACL , 508-515.
[9] Eleftherios Avramidis.2012. *Quality Estimation for Machine Translation output using linguistic analysis and decoding features*. Proceedings of the Seventh Workshop on Statistical Machine Translation, Montreal, Canada, Association for Computational Linguistics,6/2012
[10] Knight Kevin & Ishwar Chander (1994). *Automated post-editing of documents*. In Proceedings of the twelfth national
[11] R. Kneser and H. Ney. *Improved backing-off for m-gram language modeling*. In International Conference on Acoustics, Speech and Signal Processing, pages 181–184, 1995.
[12] Irina Rish. 2001.*An empirical study of the naive Bayes classifier*, IJCAI 2001 workshop on empirical methods in artificial intelligence.

Authors

Kuldeep Yogi, Assistant Professor, Department of Computer, Banasthali University, Rajasthan, India. He has worked for automatic quality estimation of machine translation outputs for manual post editing in his Ph.D. thesis and has submitted it. He developed various soft tools for automatic machine translation evaluation in a DIT granted "English Indian Language Machine Translation(EILMT)" project phase-I and is now, working for machine translation engines ranking in EILMT project phase-II. He has published several research papers in national and international journals

Chandra Kumar Jha, has been worked for 8 year as deputy manager in Indo Galf industries Ltd., Delhi. Presently, he has been working as Professor and head of Comp.Sc. Department in Banasthali University and has been coordinating Cosco CCNA course, Bachelor of vocational courses and Community College since last 14 years. He has been guiding Ph.D. scholars. More then 6 Ph.D. awarded in his guidance till now. He has published more then 52 research papers in international journals and conferences and has delivered several inviting lectures in various universities and engineering colleges. He had awarded as Best Instructor in Asian pacific region in 2011 by Cisco, USA. He was a prime investigator in a project "Digitalization of rear books" sponsored by MCIT, India.

Shivangi Dixit, Student of B.Tech,(Computer Science) II year, Banasthali University, Rajasthan