A Machine Learning Method to Distinguish Machine Translation from Human Translation

Yitong Li[1], **Rui Wang**[1,2], **Hai Zhai**[1,2] *†
[1]Center for Brain-Like Computing and Machine Intelligence,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, 200240, China
[2] Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
lrank@sjtu.edu.cn, wangrui.nlp@gmail.com, zhaohai@cs.sjtu.edu.cn

Abstract

This paper introduces a machine learning approach to distinguish machine translation texts from human texts in the sentence level automatically. In stead of traditional methods, we extract some linguistic features only from the target language side to train the prediction model and these features are independent of the source language. Our prediction model presents an indicator to measure how much a sentence generated by a machine translation system looks like a real human translation. Furthermore, the indicator can directly and effectively enhance statistical machine translation systems, which can be proved as BLEU score improvements.

1 Introduction

The translation performance of Statistical Machine Translation (SMT) systems has been improved significantly within this decade. However, it is still incomparable to the human translation (Feng et al., 2012; Li et al., 2012). Most translation text generated by SMT systems can be understood in some degree but still not good enough. However, a significant proportion of text that exists serious mistakes and even does not make sense, and these text can be easily recognized by human.

It is not difficult to understand the reason why SMT systems generate ill-formed or non-sense sentences. SMT systems combine probability models in a log-linear framework (Och and Ney, 2003), where the systems always attempt to find a sentence with the highest probability from the candidates. However, Language Model (LM), such as n-gram LM, and reordering model only have limited capacity to represent context, where sentences with local optimum could often be output. Meanwhile, it can be a very different thing for the entire translation sentence due to complicated semantic and pragmatic issues.

Therefore, to improve SMT performance, if poorly translated sentences can be distinguished automatically, it is possible for us to refine these sentences by some extra efforts. In this paper, to order to define the quality of the sentence generated by SMT systems, we borrow the idea from the evaluation of machine translation task, that the more like human translation text, the better the machine translation output is. Considering that the poorly translated sentences show great difference from human text, we compare text generated by SMT systems with human translations. This comparison motivates us to design a predictor to tell whether a sentence is machine generated or human generated. Above all, such a predictor can be treated as a binary classification problem.

In this paper, we use Support Vector Machines (SVMs) (Hearst et al., 1998) to solve such a problem. The benefits of SVMs for text categorization have been identified since it learns well with many

*Correspondence author.

†Thank all the reviewers for valuable comments and suggestions on our paper. This work was partially supported by the National Natural Science Foundation of China (No. 61170114, and No. 61272248), the National Basic Research Program of China (No. 2013CB329401), the Science and Technology Commission of Shanghai Municipality (No. 13511500200), the European Union Seventh Framework Program (No. 247619), the Cai Yuanpei Program (CSC fund 201304490199 and 201304490171), and the art and science interdiscipline funds of Shanghai Jiao Tong University, No. 14X190040031, and the Key Project of National Society Science Foundation of China, No. 15-ZDA041.

29th Pacific Asia Conference on Language, Information and Computation: Posters, pages 354 - 360
Shanghai, China, October 30 - November 1, 2015

relevant features (Joachims, 1998). In order to find those poorly SMT-translated sentences, we train an SVM-classifier on a feature space. Most features are linguistically motivated only from the target language side. As only target language is concerned, our model will be facilitated of some direct applications.

Among all features, a major part is related to the syntactic parser. The parsing structure of the output sentence is very sensible to the quality of SMT outputs. We therefore especially select these features related to the branching properties of the parse tree. One of the reason is that it had become apparent from failure analysis in (Corston-Oliver et al., 2001) that SMT system output tended to favor right-branching structures over noun compounding.

The remainder of this paper is organized as follows: In Section 2, we will give a quick review on SMT and revelent classification tasks. The SVM approaches and all the features used in our method will be presented in Section 3. Section 4 will give a description on the experiments and an analysis of corresponding results. Last, we will conclude our work in Section 5.

2 Related Work

In the classification task part, as our goal is to distinguish sentences with different quality, we are actually working on confidence estimation or automatic evaluation of SMT systems (Doddington, 2002; Papineni et al., 2002; Zhang et al., 2014).

Early work on automatic evaluation of machine translation text estimates the quality at the word level (Gandrabur and Foster, 2003; Ueffing and Ney, 2005) . Namely, n-gram features played an important role in translation quality differentiation. However, this paper considers deep level of linguistic features such as those derived from parsing tree instead of n-gram features.

Liu and Gildea (2005) also used features related to the syntactic parser. Compared with our work, they cared more about detailed syntax properties of the sentences on the parse trees. In this paper, we use less properties but more syntactic structure features.

Corston-Oliver et al. (2001) adopted parse tree related features to evaluating MT. Their work shows a high accuracy in the classification task. However, the generation of their training and test data should limit to the same SMT system. In this paper, we devote to developing a model that is capable of distinguishing texts generated by multiple sourced SMT systems from human texts. To achieve such an aim, we will introduce quite different types of features such as emotion agreement inside a sentence.

In the statistical machine translation systems part, the performance is depended on the LM and translation model. Traditional Back-off n-gram LMs (BNLMs) (Chen and Goodman, 1996; Chen and Goodman, 1999; Stolcke, 2002) have been widely used for probability estimation and BNLMs also show up in many other NLP tasks (Jia and Zhao, 2014; Zhang et al., 2012; Xu and Zhao, 2012). Recently, a better probability estimation method, Continuous-Space Language Models (CSLMs), especially Neural Network Language Models (NNLMs) (Bengio et al., 2003; Schwenk et al., 2006; Schwenk, 2007; Le et al., 2011) are being used in SMT tasks (Son et al., 2010; Son et al., 2012; Wang et al., 2013; Wang et al., 2015; Wang et al., 2014). Also, Neural Network Translation Models (NNTMs) show a success in SMT (Kalchbrenner and Blunsom, 2013; Blunsom et al., 2014; Devlin et al., 2014). However, the high cost of CSLMs makes it difficult to decoding directly. This leads to a n-best reranking method which is available for our paper (Schwenk et al., 2006; Son et al., 2012).

3 The Proposed Approach

In this Section, we present a machine learning method to distinguish poor translated sentences from good ones.

3.1 Support Vector Machine

For text classification tasks, Many approaches have been proposed (Sebastiani, 2002). Among these approaches, SVM has shown widely applications (Joachims, 1998; Joachims, 1999; Joachims, 2002; Tong and Koller, 2002). And in following subsection we will introduces how to formalize the proposed task.

The training corpus for the classifier includes l human translation sentences as positive samples and l corresponding SMT outputs as negative samples. For a sentence S, it can be represented by an

N-dimensional feature vector V $\{v_1, v_2, \cdots, v_N\}$, where N is total number of all the features, and in most cases, v_i is a real number feature normalized by the length L_S of sentence S.

With the above training corpus, we will train an SVM classifier with linear kernel. The SVM prediction function is defined as the following:

$$predict(S) = \begin{cases} +1, & h(S) \geq 0 \\ -1, & h(S) < 0 \end{cases}$$

where

$$h(S) = w_1 v_1 + w_2 v_2 + \ldots + w_N v_N$$

In this paper, Liblinear (Fan et al., 2008) is adopted as our SVM implementation and the parameter soft margin width is optimized over a small development set.

3.2 Features

In this subsection, we will present our feature collections.

Considering that only the properties of target language are involved in our expectation, we decide to use specific types of linguistic features to present the quality of the sentence.

A very important type of linguistic features is directly linked to syntactic structure of sentence. When getting the parse tree of a sentence, we can exploit a number of available properties, such as sentence structure and the densities of constituent types, to design as our features.

For parser implementation, we use Stanford Lexicalized Parser version 3.3.1. (De Marneffe et al., 2006). Figure 1 gives an example of a parse tree.

The features related to the parse tree are as the following[1]:

- number of right-branching nodes for all constituent types and for Noun Phrases (NPs).

  Using Figure 1 as an example, there are 13 right-branching nodes for all constituent types in colorful frames, including one NP in the red frame. Normalized by the length of the sentence 16, feature scores are respectively 0.8125 and 0.0625.

- number of left-branching nodes for all constituent types and for NPs
- number of pre-modifiers, adjectives before nouns, for all constituent types and for NPs
- number of post-modifiers, adjectives after nouns, for all constituent types and for NPs
- branching index, the number of right-branching nodes minus number of left-branching nodes, for all constituent types and for NPs
- branching weight index, number of tokens covered by right-branching nodes minus number of tokens covered by left-branching nodes, for all constituent types and for NPs
- modification index, the number of pre-modifiers minus the number of post-modifiers, for all constituent types and for NPs
- modification weight index, length in tokens of all pre-modifiers minus length in tokens of all post-modifiers, for all constituent types and for NPs

We also consider density of function words as well as the pronouns, where SMT systems make mistakes frequently. All densities are computed by counting the words with sentence length normalization:

- overall function word density
- density of determiners
- density of quantifiers
- density of pronouns
- density of prepositions
- density of punctuation marks
- density of auxiliary verbs
- density of conjunctions
- density of different pronoun: Wh-, 1st, 2nd, and 3rd person pronouns

[1] In default, all the following counting numbers for feature score computation are normalized by the length of the sentence.

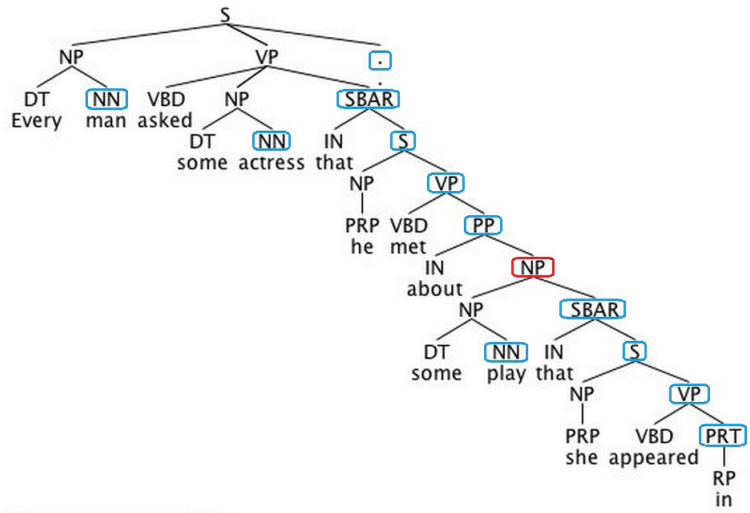

Figure 1: An Example of Parse Tree

The presence of out of vocabulary (OOV) word usually make situations more complicated. Also, problem like subject-verb disagreement are easy to be detected. Therefore, we give a group of lexical-level features:

- number of OOV words
- types of the immediate children of the root
- subject-verb disagreement

In additional, we score emotion agreement inside a sentence as features. This is motivated by the observation that a reasonable sentence should have a consistent emotion strength among different words. To evaluate such agreement, we build a dictionary $D_{emotion}$ especially for emotion words in advance, in which each word s_i can be scored from -3 to $+3$. We score all the words into these categories with a linear model to describe the strength of emotion. To a sentence, the average scoring and standard deviation will be considered:

- $\mu_{emotion}(S)$
- $\sigma_{emotion}(S)$

where S is a sentence with length len.

Finally, sizes of the following constituents are measured:

- sentence length
- parse tree depth
- maximal and average NP length
- maximal and average Adjective Phrase (ADJP) length
- maximal and average Prepositional Phrase (PP) length
- maximal and average Adverb Phrase (ADVP) length

4 Experiment

4.1 Classification

In this subsection, we will give experiment details of the prediction model.

In all of our experiments, the default settings[2] of Moses (Koehn et al., 2007) and GIZA++ (Och and Ney, 2003) are used for system building. For each SMT system, a 5-gram LM (Chen and Goodman, 1996) is trained on the target side of training set using IRST LM Toolkit.

We use four language pairs from version 7 of the Europarl corpus[3] (Koehn, 2005) as our experiment data and train four SMT systems, respec-

[2]In this paper, we build only phrase-based SMT for experiment implementation. However, we believe this method is feasible for other SMT systems, such as syntax-based SMT.

[3]http://www.statmt.org/europarl/

tively: French-English, German-English, Italian-English and Danish-English.

Considering the consistency of system and convenience of analysis, all these four systems use English as target language. We use these four systems to generate translation text.

We randomly pick 5K sentences from the French corpus, noted as $F1(5K)$, and translated into English sentences $E1(5K)$ as our negative samples, by SMT system. The corresponding English part $E1\prime$ of $F1$ is used as the positive samples. $\{E1, E1\prime\}$ forms the required training set. Then, we randomly pick 10K sentences from each of French $F2(10K)$, German $G2(10K)$, Italian $I2(10K)$ and Danish $D2(10K)$ corpora and translate them into English text $E2(40K)$. Another 40K sentences are extracted from English $E2\prime(40K)$. $\{E2, E2\prime\}$ forms a multi-model-translated-text test set. $F2$ has no cover with $F1$.

The prediction results are shown in Table 1:

Data Set	Accuracy
Training set	92.3%
Test set	74.2%

Table 1: Classification Accuracy

4.2 Feedback to SMT system

One direct application of our prediction model is to provide feedback to SMT systems.

We select the French-English SMT system that we built above as our baseline. For the sake of modifying the system as little as possible, we consider an n-best list and reranking method on the output candidates of the baseline.

We make a slight change on the prediction model so that it can give a confidence score between 0 and 1 on each sentence. The nearer with 1 its score is, the better the sentence will be. For each SMT output sentence, we choose a 1000-candidate[4] list sorted by the baseline, and score them by our prediction model. We check each candidate by the original sort, and find out the first candidate whose score is greater than a threshold H as our new output.[5] In case that no candidates satisfy the condition, we simply give the origin output.

In our experiment, we set H empirically. Table 2 shows the 1.6 BLEU score refined by our method.

MT System	BLEU Score
Baseline	23.5
Refined H = 0.6	24.7
Refined H = 0.7	**25.1**
Refined H = 0.8	23.9

Table 2: BLEU scores

4.3 Discussion

We will discuss how our method works by examples. Table 3 shows a translation and refined example.

S	Quelle que soit la bonne réponse, la question est que la détermination des mesures à prendre concernant la race représente un problème dominant dans la politique américaine.
T	Whatever the answer, the question is the determination of the action on the race is a dominant issue in American politics.
R	Whatever the answer, the question is that determining what to do about race is a dominant issue in American politics.
Ref	Regardless of the correct answer, the point is that determining what to do about race is a dominant issue in US politics.

Table 3: A Translation Example. *S*: Source, *T*: Target, *R*: Refined, and *Ref*: Reference

According to the analysis, the parse tree structure of output T is seriously right-deviated, while sentence R has a more balance tree structure. Our prediction model will consider R as a good translation but T as a bad one. When reordering candidates, our algorithm successfully selects R as output instead of T. In addition, compared with reference sentence, we see that R is an even better translation.

5 Conclusion

In this paper, we present an indicator that using linguistic features to train an SVM classifier to distinguish poor SMT sentences from good ones. We use

[4]This is an empirical value.

[5]We considered directly adding SVM score as a new feature into SMT system, however our current method shown in this paper gets better results. Also, this method is more efficient.

single-MT-model-generated text as training data and multi-MT-model-generated text as test data to show the stability of our method. With the help of a series of features derived from parse tree, emotion agreement and lexical features, our classifier gives acceptable accuracy. In addition, we show that such a predicator can effectively enhance the corresponding SMT task.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Phil Blunsom, Edward Grefenstette, Nal Kalchbrenner, et al. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.

Stanley F Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, volume 13(4):359–393(35).

Simon Corston-Oliver, Michael Gamon, and Chris Brockett. 2001. A machine learning approach to the automatic evaluation of machine translation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 148–155. Association for Computational Linguistics.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation*, volume 6, pages 449–454.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1370–1380.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Yang Feng, Dongdong Zhang, Mu Li, Ming Zhou, and Qun Liu. 2012. Hierarchical chunk-to-string translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 950–958. Association for Computational Linguistics.

Simona Gandrabur and George Foster. 2003. Confidence estimation for translation prediction. In *Proceedings of the 7th conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 95–102. Association for Computational Linguistics.

Marti A. Hearst, Susan T Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28.

Zhongye Jia and Hai Zhao. 2014. A joint graph model for pinyin-to-chinese conversion with typo correction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1512–1523.

Thorsten Joachims. 1998. *Text categorization with support vector machines: Learning with many relevant features*. Springer.

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, volume 99, pages 200–209.

Thorsten Joachims. 2002. *Learning to classify text using support vector machines: Methods, theory and algorithms*. Kluwer Academic Publishers.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation summit*, volume 5, pages 79–86. Citeseer.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011. Structured output layer neural network language model. In *2011*

IEEE International Conference on Acoustics, Speech and Signal Processing, pages 5524–5527. IEEE.

Junhui Li, Zhaopeng Tu, Guodong Zhou, and Josef van Genabith. 2012. Head-driven hierarchical phrase-based translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 33–37. Association for Computational Linguistics.

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Holger Schwenk, Daniel Dchelotte, and Jean-Luc Gauvain. 2006. Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 723–730. Association for Computational Linguistics.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech & Language*, 21(3):492–518.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Le Hai Son, Alexandre Allauzen, Guillaume Wisniewski, and François Yvon. 2010. Training continuous space language models: Some practical issues. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 778–788. Association for Computational Linguistics.

Le Hai Son, Alexandre Allauzen, and François Yvon. 2012. Continuous space translation models with neural networks. In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 39–48. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 901–904.

Simon Tong and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66.

Nicola Ueffing and Hermann Ney. 2005. Application of word-level confidence measures in interactive statistical machine translation. *In Proceedings of the 10th Annual Conference of the European Association for Machine Translation*, pages 262–270.

Rui Wang, Masao Utiyama, Isao Goto, Eiichro Sumita, Hai Zhao, and Bao-Liang Lu. 2013. Converting continuous-space language models into n-gram language models for statistical machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 845–850, Seattle, Washington, USA, October. Association for Computational Linguistics.

Rui Wang, Hai Zhao, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2014. Neural network based bilingual language model growing for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 189–195, Doha, Qatar, October. Association for Computational Linguistics.

Rui Wang, Hai Zhao, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2015. Bilingual continuous-space language model growing for statistical machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Qiongkai Xu and Hai Zhao. 2012. Using deep linguistic features for finding deceptive opinion spam. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1341–1350. Citeseer.

Xiaotian Zhang, Hai Zhao, and Cong Hui. 2012. A machine learning approach to convert ccgbank to penn treebank. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 535–542.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, and Hai Zhao. 2014. Learning hierarchical translation spans. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 183–188.