Incorporating Distinct Translation System Outputs into Statistical and Transformer Model

Mani Bansal, D.K.Lobiyal

Jawaharlal Nehru University, Hauz khas South, New Delhi, 110067, India

Abstract

To find correct translation of an input sentence in Machine Translation is not an easy task of Natural language processing (NLP). The hybridization of different translation models has been found to handle this problem in an easy way. This paper presents an approach that takes advantage of various translation models by combining their outputs with statistical machine translation (SMT) and transformer method. Firstly, we achieve Google Translator and Bing Microsoft Translator outputs as external system outputs. Then, outputs of those models are fed into SMT and Transformer. Finally, the combined output is generated by analyzing the Google Translator, Bing, SMT and Transformer output. Prior work used system combination but no such approach exist which tried to combine the statistical and transformer system with other translation system. The experimental results on English-Hindi and Hindi-English language have shown significant improvement.

Keywords

Machine Translation, Transformer, Statistical Machine Translation, Google Translator, Bing Microsoft Translator, BLEU.

1. INTRODUCTION

Machine Translation is the main area of Natural Language Processing. There are various translation approaches each with its pros and cons. One of the recent and existing approaches of Machine Translation (MT) is Statistical Machine Translation (SMT). The Statistical system is [1] structured for adequacy and handling out-of-vocabulary words. Neural Machine Translation is a breakthrough which reduces post-editing efforts [2] and helps in dealing with syntactic structure of sentence. The NMT [3] outputs more fluent translations. Therefore, we make a hybrid system by combining Statistical and Transformer (NMT with multi-head self-attention architecture) outputs to refine the machine translation outputs.

The combining these approaches into one is not an easy task. By using either SMT or Transformer does not give the solution to all issues. NMT has a problem of over-translates and under-translates to some extent. Also long distance dependency, phrase repetitions, translation adequacy for rare words and word alignment problems are observed in neural based system. As SMT [4] handles long-term dependency issues but unable to integrate the information in the source text. The additional

ISIC'21:International Semantic Intelligence Conference, February 25–27, 2021, New Delhi, India

✉ manibansal1991@gmail(M.Bansal);lobiyal@gmail.com (D.K. Lobiyal)

CEUR Workshop Proceedings http://ceur-ws.org ISSN 1613-0073 UR Workshop Proceedings(CEUR-WS.org)

information in source text helps to disambiguate the word sense, and named entity problems. The proposed architecture performed the experiment on English- Hindi and Hindi-English dataset. In that, the output of Bing Microsoft Translator and Google Translator are given as input to Statistical and Transformer model to analyze the improvement in the combined target output. If the external translator outputs are achieved by using English to Hindi (Eng-Hi) language pair, then Statistical and Transformer used the reverse language pair as input i.e. Hindi to English (Hi-Eng). Therefore, the output of external translator can be easily merged with input of other two systems i.e. Statistical and Transformer.

The paper is framed as following: In Section 2, a brief introduction of hybrid approaches proposed for Machine Translation. Section 3 elaborates our proposed approach. The experiments undertaken in this study have been discussed along with the results obtained in Section 4. In Section 5, the conclusion is presented.

2. RELATED WORK

Many methods have been presented in literature for machine translation. Researchers combined different translation techniques [5] to improve translation quality. We have identified that most of the related studies take SMT as baseline, very few studies in the literature show combination with NMT.

The Example-based, Knowledge-based and Lexical transfer system combined using chart manager in [6] and selected best group of edges with the help of chart-walk algorithm (1994). Authors in the [7] computed a consensus translation by voting on confusion network. They produced the word alignments of original machine translation hypotheses in pairs for confusion network.

Minimum Baye’s risk system combination (MBRSC) method [8] gathers the benefits of two methods- combination of sub-sequences and selection of sentences. These methods use best subsequences to generate best translation.

The lattice-based system combination model [9] entitles for phrase alignments and uses lattice to encode all candidate translations. The earlier proposed confusion network processed word-level translation whereas lattice expressed n-to-n mappings for phrase-based translation. The hybrid architecture [10], where every target hypothesis was paraphrased using various approaches to obtain fused translations for each target, and make final selection among all fused translations.

Multi-engine machine translation amalgamated output of several MT systems into single corrected translation [11]. It consists of search space, beam search decoder with its features and many accessories. As NMT decoding lacks a mechanism to guarantee all source words to be translated and favors short translations. Therefore, the authors in [12] incorporates SMT translation model and n-gram language model under log-linear framework.

3. PROPOSED APPROACH

3.1. BASIC TRANSFORMER MACHINE TRANSLATION

The Transformer model [13] accepts a source language sentence X = (x_1, x_2, ..., x_N) as an input and outputs a target language sentence Y = (y_1, y_2, ...,y_M). The NMT construct a neural network that translates X into Y by learning objective function p(Y |X) from a parallel corpus. The Transformer model is encoder-decoder model in which the encoder generates the intermediate representation h_t (t = 1, 2, ..., N) from X (source sentence) and the decoder generates Y (target sentence) from the intermediate representation h_t:

$$h_t = Transformer\ Encoder(X) \quad (1)$$

$$Y = Transformer\ Decoder\ (h_t) \quad (2)$$

The encoder and decoder are made up of stack of six layers. Each encoder layers consists of Multi-head and Feed Forward sub-layers. Whereas, each decoder layers is consists of three sub-layers. Apart from two sub-layers of encoder, decoder embeds cross-lingual multi-head attention layer for the encoder stack output.

The attention mechanisms:- Both self-attention mechanism and cross-lingual attention are computed as follows:

$$\textit{Attention}\ (\mathrm{Q, K, V}) = \textit{softmax}\ (\frac{QK^T}{\sqrt{d_{model}}})\ \mathrm{V} \quad (3)$$

Here, Q represent Query vector, V and K represent as Value and Key vector of both encoder and decoder respectively and $\mathrm{d_{model}}$ is the size of this key vector. The product of Query and Key represents the similarity between each element of Q and K and it is converted to a probability by using the softmax function, which can be treated as weights of attention of Q to K.

The self-attention captures the degree of association between words in the input sentence by using the Q, K and V in the encoder. In similar way, the self-attention in the decoder captures the degree of association between words of output sentence by using Query, Key and Value in the decoder. The cross-lingual attention mechanism computes the degree of association between a word in source and target language sentence by using Query of decoder and output of last layer of encoder as Key and Value. In the multiple head self-attention with h number of heads, Q, K, and V are linearly projected to h subspaces, and then the attention function is used in parallel on each subspace. The concatenation of these heads is projected to a space with the original dimension.

$$\textit{MultiHead}\ (\mathrm{Q, K, V}) = \textit{Concat}\ (\textit{head}_1, \ldots., \textit{head}_h)\mathrm{W^O} \quad (4)$$

$$\textit{head}_i = \textit{Attention}\ (\mathrm{Q}\ W_i^Q, \mathrm{K}W_i^K, \mathrm{V}W_i^V) \quad (5)$$

where, $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $\mathrm{W^O} \in \mathbb{R}^{hd_v \times d_{model}}$, are weight matrices and *Concat* is a function that concatenates two matrices. Multiple head attention learns information from representation spaces at different positions. The Transformer uses position encoding (PE) to encode the position related information of each word in a sentence because the Transformer does not have any recurrent or convolution structure. PE is calculated as follows:

$$\mathrm{PE\ (pos, 2i) = sin\ (pos/10000}^{2i/d_{model}}) \quad (6)$$

$$\mathrm{PE\ (pos, 2i + 1) = cos(pos/10000}^{2i/d_{model}}) \quad (7)$$

where, *i* is dimension or size, and *pos* is absolute position of the word.

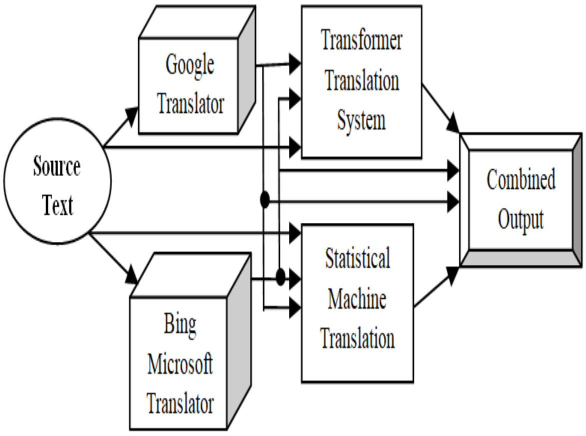
Google Translator
Transformer Translation System
Source Text
Combined Output
Statistical Machine Translation
Bing Microsoft Translator

Fig 1. Translation System Architecture

3.2. COMBINING MACHINE TRANSLATION OUTPUT WITH TRANFORMER MODEL

The transformer system inputs are the translated outputs of different or external translation methods and same source sentence as shown in Fig 1. Then, we used three encoders: one for Google output text, another uses Bing and source language encoder. The concatenation generated by the three encoders is fed into conventional Transformer decoder.

The Google output text encoder represented as $X_1 = (x_{11}, x_{12}, ..., x_{1N})$ input. The Bing output sentence represented as $X_2 = (x_{21}, x_{22}, ..., x_{2N})$ and third transformer encoder accepts a source language sentence $X_3 = (x_{31}, x_{32}, ..., x_{3N})$ as an input. Then, Transformer encoder generates the intermediate representation h_g (t = 11,...., 1N), h_b (p = 21,...., 2N), h_t (t = 31,...., 3N), from the source language sentence X_1, X_2 and X_3. The intermediate representation h_g, h_b, h_t:

$$h_g = Transformer\ Encoder(X_1) \quad (8)$$

$$h_b = Transformer\ Encoder(X_2) \quad (9)$$

$$h_t = Transformer\ Encoder(X_3) \quad (10)$$

After the intermediate representations of inputs, these are concatenated in the composition layer [14]. The concatenation *h* is the output of the proposed encoder and is fed to the decoder of our model.

$$h = Concat\ (h_g, h_b, h_t) \quad (11)$$

The decoder generated the combined target language output using the above expressions.

3.3. BASIC STATISTICAL MACHINE TRANSLATION

The phrase translation method or Baye's Rule forms the basis of Statistical Translation [15]. The best translation output sentence e_{best} is formulated as follows:

$$e_{best} = \text{argmax}_e\ P(e|f) = \text{argmax}_e\ [P(f|e)\ P_{LM}(e)] \quad (12)$$

where, f is source sentence and e is target sentence. $P_{LM}(e)$ and P(f |e) are language model (LM) and the translation model (TM) , respectively. The input text *f* is partitioned uniformly into a sequence of T phrases $\bar{f}_1^T$. Each $\bar{f}_t$ foreign phrase in $\overline{f_1^T}$ is translated into $\bar{e}_t$ english phrase. The translation model P(f|e) is disintegrated into:

$$P(\bar{f}_1^T|\bar{e}_1^T) = \prod_{i=1}^{T} \phi\ (\bar{f}_t|\bar{e}_t)\ \ d\ (\alpha_i - \beta_{i-1}) \quad (13)$$

The phrase translation is formed by probability distribution ϕ. The relative distortion probability distribution d (α_i – β_{i-1}) calculates the output phrases.

3.4. COMBINING MACHINE TRANSLATION OUTPUT WITH STATISTICAL SYSTEM

The Statistical combination approach uses three modules: Alignment module, Decoding and Scoring. The alignment is useful for string alignments of the hypotheses generated from different machine translation systems. A decoding step builds hypotheses using aligned strings from previous step by using beam search algorithm. The final scoring step helps in estimating the final hypotheses.

3.4.1. ALIGNMENT

The single best outputs d_1, d_2, ...d_m from each of the m participating systems are taken into consideration. We take sentence pairs d_i and d_j, and strings between the sentence pairs are aligned. For m sentences, $\frac{m(m-1)}{2}$ possible sentence pairs are required to be aligned. The string w_1 in sentence d_i aligned to string w_2 in sentence d_j following two conditions:- Firstly, w_1 and w_2 are same. Then, w_1 and w_2 have Wu and PaLMer [16] similarity score > δ. METEOR [17] is used to align sentences configuration.

3.4.2. DECODING

In decoding, best outputs are combined of participating systems to form a set of hypothesis. The first word of a sentence is used to start the hypothesis. At any moment of time, the search can be shifted to a different sentence or addition of the new words continued using words from the previous sentence. Let a word w is added to the hypothesis, taken from the best output d_i or shift to different output d_j. On shifting, the first left over word from best output sentence is added to next hypothesis. With the help of this method, a hypothesis can be made using various system outputs. If a hypothesis made up of at most single word from each set of aligned words, there is less possibility of occurrence of duplication.

The search space is easily switched through sentences, and thus maintaining adequacy and fluency is difficult. Therefore, hypothesis length, language model probability and number of n-gram matching between individual system's output and hypothesis, features are used for complete hypothesis.

3.4.3. BEAM SEARCH

In this search, the number of equal length hypotheses is assigned to beam. The hypotheses are recombined by feature state, history of the hypothesis appropriate to the length requested by features and search space hashing. Then, pointers are maintained of recombined hypotheses that are packed into single hypothesis. Therefore, it enabled extraction of k-best.

3.4.4. SCORING

In the output of decoding steps, the m-best lists are generated. Language Model Probability [18] and Word Mover's Distance [19] methods are used to calculate scoring of m-best list. The m-best list is represented as $h_1, h_2, h_3, \ldots., h_p$ and the score of each h_i is calculated. The minimum error rate [20] training (MERT) method is used to calculate the weights.

4. EXPERIMENTATION AND RESULTS

The proposed approach is tested on HindEnCorp [21] dataset. It contains 273,880 sentences. For preparing training and development set, we use 272,880 (267,880 for training + 5k for tuning) sentences for statistical system and transformer. The test set contains 1k sentences. The output of Google Translate[1] and Bing Microsoft Translate[2] is combined with SMT and transformer. Our proposed architecture should be trained along with the outputs of various translated sentences.

We trained and tested our approach on one more dataset from ILCC (Institute for Language, Cognition and Computation) for English to Hindi language which contains 43,396 sentences. For Hindi to English translation, we used TDIL-LC (Indian Language Technology Proliferation and Deployment Centre) dataset divided into tourism (25k sentences) and health (25k sentences) domain. Therefore, Hindi to English language pair trained and tested on 50k sentences.

We train Statistical Machine Translation with KneserNey smoothing [22] for probability distribution of 4-gram language model (LM) by using IRSTLM [23]. The Moses decoder [24] finds highest scoring sentence for phrase-based system. The model learns the heuristics using GIZA++ [26] and word alignment with gro-diag-final.

The Transformer[3] system contains encoder and decoder six layers, eight attention heads, and 2048 feed-forward inner-layer size or dimensions with dropout = 0.1. The hidden state and word embedding dimension d_{model} is 512. We limit maximum sentence length to 256, and input and the output tokens per batch are limited to 2048. We used Adam optimizer [26] with $\beta_1 = 0.90$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$.

[1] https://translate.google.com/

[2] https://www.bing.com/translate

[3] https://github.com/tensorflow/tensor2tensor

Further, we used length penalty α = 0.6 and beam search with a size of 4.

The Bilingual Evaluation Understudy (BLEU) [27] selected as primary evaluation algorithm. It evaluates the quality of machine translated text with that of human translation. Scores in BLEU are calculated for translated sentences by comparing with good quality reference sentences. The scores are normalized over the complete dataset to estimate overall quality. BLEU calculates the score always between 0 and 1, but it shows the score in percentage form. If BLEU score are more close to 1 better is the accuracy of the system.

The results on different test sets are obtained for Hindi to English (Hi-Eng) and English-Hindi (Eng-Hi) language pairs. It is evident from Table1 that translation system combination shows better results than individual system i.e. SMT with Google and Bing improved approximately 4 bleu scores than SMT alone in all language pairs.

Table 1:
Translation results with respect to BLEU score of multiple methods using different dataset.

	BLEU Score (HindiEnCorp)		**BLEU Score**	
Models	**Eng-Hi**	**Hi-Eng**	**Eng-Hi** (ILCC)	**Hi-Eng** (TDIL-DC)
SMT	9.41	8.03	7.88	7.14
Transformer	12.52	11.89	8.36	9.33
Google	11.20	10.42	8.14	7.56
Bing	11.92	10.61	9.65	8.07
SMT + Google + Bing	**15.37**	**14.70**	**11.92**	**10.82**
Transformer+ Google + Bing	14.36	13.85	11.04	10.51
SMT + Transformer + Google + Bing	13.09	12.66	10.16	9.29

The output from individual system contains some erroneous or un-translated words. But the selection of best phrase among different translated outputs (Google and Bing) generated by participating systems makes the target sentence more accurate. We also observe in the Table1 that by increasing number of MT system does not help in improving accuracy i.e. Bleu scores of SMT, Transformer, Google and Bing together achieved 2 points less than SMT, Google and Bing. The BLEU score achieved by using SMT, Bing Microsoft and Google translator together are highest. Also the scores of Transformer, Google and Bing are better than using all translation models and bleu scores improved by 1 point. The scores retrieved using TDIL-DC and ILCC dataset are lesser than HindiEnCorp because the size of dataset is very less. The overall accuracy of our translation output using reverse language pair is improved by combining the better parts of outputs. But, the Bleu scores are not improved much in our approach. The main reason is that the error occurred in external machine translation systems, would also reflect in the combination approach. Hence, by removing these errors, we will try to achieve better results in future.

5. CONCLUSION

We investigated the approach of combining the various translation outputs with statistical machine translation and Transformer which improve the final translation in this paper. The proposed method increased the complexity of the overall system. Experimentation on Hindi to English and from English to Hindi shows that incorporating different system output achieves better result than individual system. In future, we extend this approach for translation of other language pairs and tasks like text abstraction, sentence compression. We will also try to incorporate BERT model into the neural based English to Hindi translation and will also explore the graph based encoder-decoder translation methods.

REFERENCES

[1] D. Jiang, A Statistical Translation Approach by Network Model, in: Recent Developments in Intelligent Computing, Communication and Devices, Springer, Singapore (2019), pp. 325-331.

[2] A. Toral, Martijn Wieling, and Andy Way. Post-editing effort of a novel with statistical and neural machine translation. Frontiers in Digital Humanities, 5.9(2018). doi:10.3389/fdigh.2018.00009.

[3] L. Zhou, W. Hu, J. Zhang, C. Zong, Neural System Combination for Machine Translation. arXiv preprint arXiv:1704.06393 (2017).doi: 10.18653/v1/P17-2060

[4] W.He, Z. He, H. Wu, H. Wang, Improved Neural Machine Translation with SMT Features, in: Thirtieth AAAI conference on artificial intelligence (2016).

[5] S. Attri, T. Prasad, G. Ramakrishna, G, HiPHET: A Hybrid Approach to Translate Code Mixed Language (Hinglish) to Pure Languages (Hindi and English). Computer Science, 21.3 (2020). doi: 10.7494/csci.2020.21.3.3624.

[6] S. Nirenburg, R. Frederking, Toward Multi-Engine Machine Translation. In Human Language Technology: Proceedings of a Workshop, Plainsboro, New Jersey (1994).

[7] E. Matusov, U. Nicola, Computing Consensus Translation for Multiple Machine Translation Systems using Enhanced Hypothesis Alignment. in: 11th Conference of the European Chapter of the Association for Computational Linguistics, (2006).

[8] J. González-Rubio, C. Francisco, Minimum Bayes' Risk Subsequence Combination for Machine Translation. Pattern Analysis and Applications (2015) 523-533. doi: 10.1007/s10044-014-0387-5.

[9] Y. Feng, Y. Liu, H. Mi, Q. Liu, Y. Lu, Lattice-Based System Combination for Statistical Machine Translation. in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009, pp. 1105-1113.

[10] W.Y. Ma, K. McKeown, System Combination for Machine Translation through Paraphrasing. in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1053-1058. doi: 10.18653/v1/D15-1122.

[11] J. Zhu, M. Yang, S. Li, T. Zhao, Sentence-Level Paraphrasing for Machine Translation System Combination. in: International Conference of Pioneering Computer Scientists, Engineers and Educators, Springer, Springer, Singapore, 2016, pp. 612-620. doi: 10.1007/978-981-10-2053-7_54.

[12] K. Heafield, A. Lavie, Combining Machine Translation Output with Open Source: The Carnegie Mellon Multi-Engine Machine Translation Scheme. The Prague Bulletin of Mathematical Linguistics, 2010, pp. 27-36. doi:10.2478/v10108-010-0008-4.

[13] A. Vaswani, N. Shazeer, J. Parmar, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is All You Need. in: Advances in Neural Information Processing Systems, 2017, pp. 5998-6008.

[14] A. Currey, K. Heafield, Incorporating source syntax into transformer-based neural machine translation. in: Proceedings of the Fourth Conference on Machine Translation, vol.1, 2019, pp. 24-33.

[15] P. Koehn, Statistical machine translation. Cambridge University Press, 2009.

[16] Z. Wu, M. Palmer Verb Semantics and Lexical Selection. arXiv preprint cmp-lg/9406033 (1994).doi: 10.3115/981732.981751

[17] S. Banerjee, A. Lavie, METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. in: Proceedings of the Acl Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine

Translation and/or Summarization, 2005, pp. 65-72.

[18] T. Brants, A.C. Popat, P. Xu, F.J. Och, J.Dean, Large Language Models in Machine Translation (2007).

[19] M. Kusner, Y. Sun, N. Kolkin, K.Weinberger, From Word Embeddings to Document Distances. in: International Conference on Machine Learning, 2015, pp. 957-966.

[20] O. Zaidan, Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems. The Prague Bulletin of Mathematical Linguistics, 91.1 (2009): 79-88. doi:10.2478/v10108-009-0018-2.

[21] O. Bojar, V. Diatka, P. Rychlỳ, P. Stranák, V. Suchomel, A. Tamchyna, D. Zeman, Hindencorp-Hindi-English and Hindi-only Corpus for Machine Translation. in: Proceedings of the 9th International Conference on Language Resources and Evaluation, 2014, pp. 3550-3555.

[22] R. Kneser, H. Ney, Improved Backing-off for m-gram Language Modeling. In: 1995 International Conference on Acoustics, Speech, and Signal Processing, IEEE, 1995, vol. 1, pp.181-184. doi: 10.1109/ICASSP.1995.479394

[23] M. Federico, N. Bertoldi, M. Cettolo, IRSTLM: An Open Source Toolkit for Handling Large Scale Language Models. in: Ninth Annual Conference of the International Speech Communication Association, 2008.

[24] H. Hoang, P. Koehn, Design of the Moses Decoder for Statistical Machine Translation. in: Proceedings of Software Engineering, Testing, and Quality Assurance for Natural Language Processing, 2008, pp. 58-65.

[25] F. J. Och, H. Ney, A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29.1 (2003), 19-51.

[26] D.P. Kingma, B.J. Adam, A method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2015).

[27] K. Papineni, S. Roukos, T. Ward, W. Zhu, BLEU: a method for automatic evaluation of machine translation. in: Proceedings of the 40th annual meeting on association for computational linguistics, 2002, pp.311-318. doi: 10.3115/1073083.1073135.