Bilingual language translation by learning

Sudharshan T R[1], Dr. Ashwini M Joshi[2]

Dept of CSE, PES University Bengaluru

[1]pes1ug19cs534@pesu.pes.edu

[2]ashwinimjoshi@pes.edu

Abstract— **This paper aims to democratize bilingual machine translation by providing an architecture which can use existing models trained on mono-lingual corpora and requiring minimal fine-tuning time. This is achieved by deviating from the standard encode-attend-decode approach by learning a translation which has the same meaning and likelihood as the given input sentence rather than generating a target sentence which has a high likelihood as the given sentence. To compute the meaning of a sentence, an encoder model like BERT is used and to compute the probability of a sentence, a causal language model like GPT is used. Suppose to translate from English to French, 4 models namely eng_encoder, eng_lm, fr_enoder and fr_lm is necessary with en_encoder and fr_encoder fine-tuned to have parallel sentences have same meaning vectors. Then for a given English sentence, compute its meaning vector and its likelihood, learn a French sentence with same meaning vector and likelihood.**

***Keywords*—— Multi-lingual neural machine translation, natural language processing, transformer, deep learning**

I. Introduction

Machine translation is a vibrant area of research with then state-of-the-art models proposed as recent as 2014 becoming ancient. With the advent of transformers[2], models high quality translations have become a reality. But these models contain millions of parameters and training them from scratch (for a new language pair) requires high computational resources which very few have an access to. In many other NLP tasks, transformer-based models like BERT[3] and GPT[16] come to the rescue as they contain vital language related representations as a result of their mono-lingual pre-training. To use them in a specific NLP task, just fine-tune the pre-trained model on task-specific dataset. This approach while highly applicable in other NLP tasks, using them for machine translation is hard which makes training NMT models for language pair a computational nightmare. This paper provides a way to use pretrained models on monolingual data effectively to translate sentences from a desired source to target language with very minimal fine-tuning. This is achieved by deviating from the standard encode-attend-decode approach by *learning* a

translation which has the same *meaning* and *likelihood* as the given input sentence rather than generating a target sentence which has a high likelihood as the given sentence.

To compute the meaning of a sentence, an encoder model like BERT[3] is used and to compute the probability of a sentence, a causal language model like GPT[16] is used. Suppose to translate from English to French, 4 models namely eng_encoder, eng_lm, fr_enoder and fr_lm is necessary with en_encoder and fr_encoder fine-tuned to have parallel sentences have same meaning vectors. Then for a given English sentence, compute its meaning vector and its likelihood, learn a French sentence with same meaning vector and likelihood.

II. Problem Statement

The main idea is to learn a target translation which has a similar meaning and likelihood of occurrence as the given input sentence.

More formally, given an input sentence *s*, learn a sentence *t* such that:

$$t = argmin_t \; L(s,t) \text{ - (1)}$$

Where *L(s, t)* is the loss function defined below:

$$L(s,t) = \|M(s) - M(t)\| + |ln(P(s)) - ln(P(t))| \text{ - (2)}$$

Where:

M(x) is the meaning vector of a sentence x,
P(x) is the probability of a sentence x.

III. Proposed Architecture

This section shall cover the various aspects of the proposed architecture and implementation considerations.

A. Encoding a sentence/embedding sequence into a meaning vector

Given a sequence of tokens (sentence) or a sequence of embedding vectors compute its meaning (a vector) using an off-the-shelf encoder model.

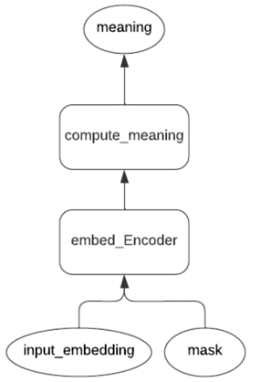

Fig 1: *get_meaning_from_embeddings*: Computes meaning vector of a sequence of input_embeddings

Given a sequence of *input_embeddings*, the *embed_encoder* computes a sequence of embeddings with contextual information incorporated. From the sequence of embeddings, *compute_meaning* computes the meaning vector from these embeddings. In this paper, I incorporate a simple average function over the *output_embeddings* which is also what has been used in sentence similarity problems

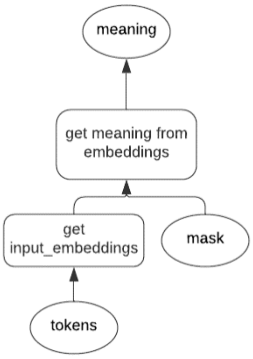

Fig 2: Get meaning from tokens

The *get input_embeddings* yields the embedding of a token without positional information encoded (as this will be handled in the *embed_encoder*).

B. Obtaining probability of a sequence of tokens/embeddings:

Given a sequence of tokens, to compute its log prob using an off-the-shelf causal language model

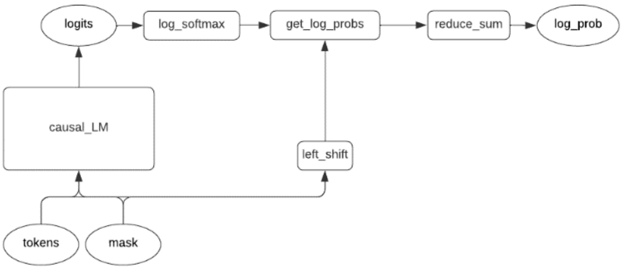

Fig 3: Computing *log_prob* of a sequence of tokens

Computing log prob for a sequence of tokens is straightforward (just lookup the probability of the next token) as causal language models are designed to do so. However, computing log prob for a sequence of embeddings is tricky as we do not actually know the tokens corresponding to the given embeddings. Hence, probabilities of a given embedding corresponding to a given token is computed and a weighted sum (instead of the traditional lookup) over the *log_softmax* outputs is calculated as the *log_prob* of an embedding.

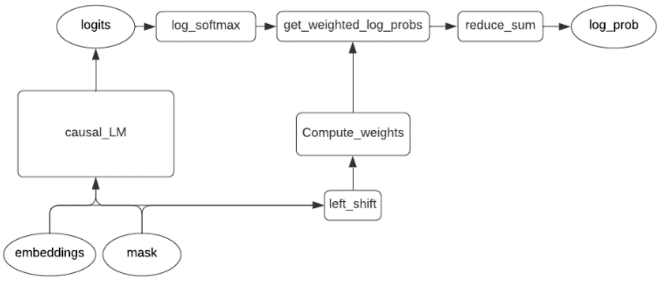

Fig 4: Computing log prob of a sequence of embeddings

C. The Adapter layer:

The causal LM and the encoder have different vocabularies and different embeddings for the tokens in common. We would need to have an adapter layer that can convert an embedding in the encoder domain to one in causal LM domain. The adapter layer is basically a dense neural network with residual connections to mitigate the vanishing gradient problem.

D. How translation happens:

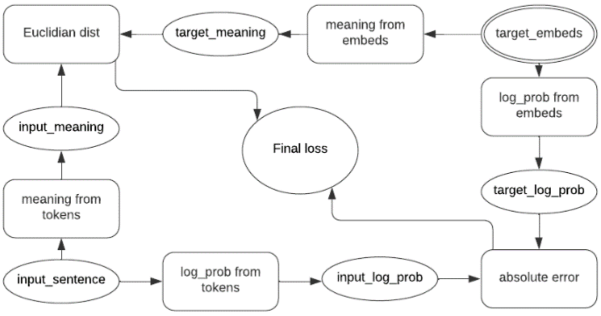

Fig 5: Language translation model architecture: The target_embeds continuously change to decrease final loss

Given an input sentence, its log prob and meaning is computed. An initial sequence of random embeddings (called target embeddings) is set and its log prob and meaning is computed. Given the log probs and meaning of the input sentence and the sequence of embeddings, the error is computed using the loss function defined in chapter 2. This error is minimized by updating the target embeddings using gradient descent. The mask for the sequence of target embeddings is already pre-computed (as a function of the length of input tokens).

IV. RESULTS AND DISCUSSIONS

Why does the problem statement defined in chapter 2 achieve the goal of machine translation?

It is known that word distributions in the real world follow the power law (number of words having a probability of occurrence p is inversely proportional to p).

Consider a graph of words being nodes and a weighted edge between 2 words v1 and v2 indicates number of times v1 and v2 are adjacent. This graph is known to have the following properties:

- High clustering coefficient (words clump together to form small clusters)
- Follows power law (few vertices have a high weighted degree)
- Large component (about 80% of words are all connected): This indicates that the many small dense components are connected through the few high degree vertices to maintain the high clustering coefficient.

This indicates that even though there are many words with low occurring probabilities, these words are mostly not related (part of many different clusters) and therefore will have different meanings.

If the same assumptions are made for sentence distributions (few sentences have a high probability of occurrence), the log_prob loss reduces the sentence search space to those sentences with same probability as the source sentence and irrespective of the size of the reduced search space very few sentences (part of the same cluster of the source sentence) will have the same meaning as the source sentence any of which is a valid translation.

Hence, if a candidate sentence minimizes both losses, it is guaranteed to be a sentence from the same cluster as the source sentence which is known to be tight nit.

V. CONCLUSION AND FUTURE WORK

In this paper, I have presented a novel way to perform bilingual language translation through learning the most plausible target sentence using encoders and language models which unlike the traditional encoder and decoder are not coupled together. This allows us to use a much wider variety of pretrained models requiring minimal fine-tuning, which makes bilingual language translation training a lot more accessible. I plan on implementing this paper for languages with high quality encoders and language models and then extending this for low resource languages.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Dr. Shylaja M Sharath, Chairperson of the Department of Computer Science and Engineering at PES University, for her encouragement and support throughout my research. I would also like to thank Dr. H B Mahesh, Capstone Coordinator, for coordinating my capstone project which produced this paper. I would also like to thank the panel members and my friends who gave me valuable feedback.

REFERENCES

[1] Mike Schuster and Kaisuke Nakajima. 2012. “Japanese and Korean voice search”. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP’12). IEEE, 5149–5152

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. “Attention is all you need”. Advances in neural information processing systems, 30.

[3] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. “Bert: Pre-training of deep bidirectional transformers for language understanding”. arXiv preprint arXiv:1810.04805.

[4] Firat, O., Cho, K. and Bengio, Y., 2016. “Multi-way, multilingual neural machine translation with a shared attention mechanism”. arXiv preprint arXiv:1601.01073.

[5] Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G. and Hughes, M., 2017. “Google’s multilingual neural machine translation system: Enabling zero-shot translation”. Transactions of the Association for Computational Linguistics, 5, pp.339-351.

[6] Sennrich, R., Haddow, B. and Birch, A., 2015. “Neural machine translation of rare words with subword units”. arXiv preprint arXiv:1508.07909.

[7] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K. and Klingner, J., 2016. “Google's neural machine translation system: Bridging the gap between human and machine translation”. arXiv preprint arXiv:1609.08144.

[8] Dabre, R., Chu, C. and Kunchukuttan, A., 2020. “A survey of multilingual neural machine translation”. ACM Computing Surveys (CSUR), 53(5), pp.1-38.

[9] Doddapaneni, S., Ramesh, G., Kunchukuttan, A., Kumar, P. and Khapra, M.M., 2021. “A primer on pretrained multilingual language models”. arXiv preprint arXiv:2107.00676.

[10] Aharoni, R., Johnson, M. and Firat, O., 2019. “Massively multilingual neural machine translation”. arXiv preprint arXiv:1903.00089.

[11] Hochreiter, S. & Schmidhuber, J"urgen, 1997. “Long short-term memory”. Neural computation, 9(8), pp.1735–1780.

[12] Cettolo, M., Girardi, C. and Federico, M., 2012. “Wit3: Web inventory of transcribed and translated talks”. In Conference of european association for machine translation (pp. 261-268).

[13] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A. and Raffel, C., 2020. “mT5: A massively multilingual pre-trained text-to-text transformer”. arXiv preprint arXiv:2010.11934.

[14] Pires, T., Schlinger, E. and Garrette, D., 2019. “How multilingual is multilingual BERT?”. arXiv preprint arXiv:1906.01502.

[15] Zoph, B., Yuret, D., May, J. and Knight, K., 2016. “Transfer learning for low-resource neural machine translation”. arXiv:1604.02201.

[16] https://paperswithcode.com/method/gpt