Large Language Models in Machine Translation - A Critical Review

G04 - RA 1

1 Summary

This paper addresses the central problem of machine translation in the context of language models; given any source language text, produce its translation in a target language, such as French to English. The paper primarily discusses scalability issues in n-gram language models (both model and data size) and proposes mechanisms to deal with them.
One issue is the difficulty of computing the probabilities for smoothing, for which the paper introduces a new mechanism, 'Stupid Backoff'. The main advantage of this is that it does not use any discounting methods but instead uses the counts of the tokens. This allows us to use a distributed training environment to compute the probabilities, which is both cheaper and produces similar results as Kneser-Ney smoothing when dealing with large data sets.

The paper uses the MapReduce programming model to generate language models based on the training data. This is done in 3 inter-connected stages:

- **Vocabulary generation**
  This assigns IDs to terms based on frequency in the training text for efficient storage. Terms that appear less often than a set threshold are mapped to a special ID representing an unknown word.
- **Generation of n-grams**
  The process of n-gram generation involves first converting words into IDs. Then, n-grams up to a certain order are emitted. A shard function places the values required for the numerator and denominator of relative frequencies into the same shard. Exceptions are made for unigrams, where all shards receive a count of the total number of words.
- **Language model generation**
  Here, n-grams and their counts are processed. The system uses a sharding function to place the values required for the numerator and denominator of relative frequencies in the same shard. All shards receive unigram counts to allow for relative frequencies to be calculated within the shards.

To integrate the language model into a distributed training system, the authors propose a new decoder architecture that utilises batch processing to improve efficiency. The decoder queues n-gram requests and sends them to servers in batches to minimise network overhead. The results of the experiments conducted on several models with varying token inputs show that increasing the amount of training data leads to almost linear improvements in BLEU score.

2 An Overview of Strengths and Weaknesses

2.1 Strengths

The paper has a series of strengths that span across several fronts. Notably, these strengths manifest in the originality of ideas, articulation of challenges and a deeper presentation of facts and figures beyond the cursory layer of presentation.

Firstly, the authors demonstrate commendable originality in introducing the Stupid Backoff as a less expensive smoothing method. Moreover, they also propose a set of ideas for improving the distributed training of language models. One such idea is to store the smoothed n-gram probabilities instead of raw sub-corpus counts in the workers. This departure from convection enables them to incorporate a single-pass decoding with their fully distributed model instead of relying on a usual two-pass approach (Ney and Ortmanns, 1999). This departure from established practices showcases the authors' ingenuity in addressing efficiency and overhead concerns which is a major strength of the paper.

Secondly, the authors' ability to clearly articulate the challenges they faced when training a model at this scale (up to a trillion tokens) sets the stage perfectly for their proposed Stupid Backoff method. For example, the number of additional steps required for the distributed training of the Kneser-Ney algorithm (Table 1, page 5) already makes it a daunting option compared to the much simpler Stupid Backoff method.

Thirdly, the authors' ability to go beyond just stating facts or equations adds more meaning to their work. We would like to cite two distinct examples of this:

- When presenting the stupid backoff model formally (Brants et al., 2007) (equation 5, page 2), the authors articulate that this equation has shifted the paradigm outside the domain of legal probabilities due to the lack of any normalization scheme. Furthermore, they explain why this lack of normalization would not affect results by linking to an earlier equation, which suggests that the optimization task of language modeling depends on relative rather than absolute feature-function values.

- While presenting the results, the authors delve deeper into the nitty gritty details of the perplexity score and provide reasons as to why their perplexity is higher than expected while testing (Brants et al., 2007) (data representation issue, section 7.3, page 8). Moreover, they also mention that perplexity itself cannot be a good benchmark for comparing models of different sizes. This transparency helps the readers weigh the results more carefully rather than blindly trusting the graphs on display.

2.2 Weaknesses

While the paper excels overall, there are occasions where the presentation of ideas could have been better. These weaknesses manifest in two major branches: use of voodoo constant and neglect to describe the evaluation challenges.

In section 4 (Brants et al., 2007), while presenting their chosen alpha value (section 4, page 3), the authors suggest that it could be made to depend on the n-gram size; however, they do not articulate why they chose a heuristic value of 0.4 particularly. This gives off the impression of a voodoo constant, which is not uncommon in machine learning (Vaswani et al., 2017) (for the number of attention heads for example). However, it would have been better had the authors cited a few empirical observations that drove their choice.

Additionally, the authors describe a myriad of shortcomings in the Perplexity metric for judging their models, yet they persist with it (section 7.3, page 8). There must have been a reason that compelled the authors to stick with the metric, but they failed to mention it. Although the task of evaluating language models is an entirely different research niche, it would have added immensely to the paper had the authors dedicated a subsection to describe the massive challenge that language model evaluation actually is rather than just citing shortcomings of Perplexity and sticking with it.

Despite these weaknesses, the significance of the paper is undeniable. The ability of Stupid Backoff to outperform the SOTA Kneser-Ney smoothing alluded to the future of NLP and machine learning in general where large models and even larger datasets would reign supreme. The AlexNet paper (Krizhevsky et al., 2012) would also use a distributed training protocol and a cheaper activation (ReLU) to revolutionize Computer Vision and kick-start the deep learning era as we know it.

3 A List of Pressing Questions

Upon reading the paper, a myriad questions came to us, answering which we think would lead to greater inroads in the topic and perhaps enlighten the readers further. The following is the formalised list of the questions:

- **What, if any, preprocessing steps were taken?**

When considering how the datasets were used to generate the vocabulary and n-grams, one has to be sure what the data looked like schematically. As such, including the preprocessing steps, if any, would be helpful.

- **Why did they choose an alpha value of 0.4 for all the experiments?**
  In footnote 2, the authors comment, 'The value of 0.4 was chosen empirically based on good results in earlier experiments.' For any hyperparameters being chosen in the experiment, it would be better if the authors provided some insight into the experiment's results that led to them selecting the alpha value rather than having users blindly believe in it.

- **What were the energy costs of the training?**
  To better understand the power efficiency of the model, precise statistics would be welcome. It is not enough to provide the number of machines and duration of training.

- **Could a smaller frequency cutoff for the vocabulary lead to a greater reduction in the BLEU score?**
  We observe that the increase in BLEU score is the slowest for the web dataset, which has the highest frequency cutoff for inclusion in the vocabulary (200). It would be interesting to see if reducing the frequency leads to a better trend in the BLEU score.

- **How does the model order affect the BLEU score trend?**
  In footnote 10, the authors mention that the order in which the models are combined causes variations in the trend of the BLEU score increase. This discussion has been omitted due to space limitations, but we want to see it. We think including the discussion could be very beneficial as it would help clarify the relationship between the BLEU score trend and dataset size, frequency-cutoff vocabulary construction, and model combination.

- **Does a relationship exist between the increase in n-gram coverage and dataset size?**
  The authors mention that the n-gram coverage increases similarly when doubling each dataset's size. We think this is an interesting correlation to observe and wonder if there exists a relationship that can be (somewhat) formally defined. Doing so would allow the construction of deliberate, lean datasets that offer the best balance between n-gram coverage and size.

4 A Discussion on Observed Limitations

Below we have compiled a section-wise overview of a few limitations that were either latently present in the work or were explicitly discussed by the authors. We are appreciative of the honesty on display by the authors throughtout the work (apprent in Results section when talking of the Perplexity metric). Thus, we have listed a few suggestions/improvements to facilitate a healthy review process.

4.1 Related Work on Distributed Language Models

The authors discuss prior work and mention certain limitations associated with that approach. The original methodology indicates a two-step decoding process involving communication with multiple distributed systems to generate translations. However, the authors propose an alternative strategy where they only have to contact one system. This optimization is achieved by maintaining pre-calculated probabilities for various phrases rather than relying on raw data. Their approach demonstrates an efficiency advantage.

4.2 Stupid Backoff

Stupid backoff does not generate probabilities but rather scores. This could be a potential limitation because many applications using language models rely on normalized probabilities to make decisions. We suggest the authors discuss this as a limitation and propose a workaround for users who want to use probabilities rather than scores.

4.3 Generation of n-grams

The text mentions that sharding based solely on the first word of n-grams can lead to imbalanced shards, impacting runtime. The authors need to mention this as a potential limitation and elaborate on how they fixed it (using the first two words for hashing).

4.4 Distributed Application

- Network Latency
  The authors correctly discuss the challenge of

network latency in a distributed system, where accessing n-grams from servers is slower than having them locally—a limitation for latency-sensitive applications.

- Perplexity Limitations
  The text mentions perplexity as a measure of language model quality but acknowledges limitations due to the mismatch between training and test data genres (news articles vs. mixed genres). This suggests perplexity might not be the most reliable metric here.

- Limited Comparison between Smoothing Methods
  The experiment uses fixed backoff factors for Stupid Backoff, while Kneser-Ney smoothing uses optimized factors. This limits a direct comparison between the two methods.

- Focus on BLEU Score
  The experiment relies on BLEU score for machine translation evaluation. While BLEU is a common metric, it has limitations in reflecting actual human translation quality. For example, BLEU evaluates translations at the sentence level and does not consider semantic meaning or coherence. Secondly, BLEU penalizes translations for words that do not appear in the reference translations, which can be problematic for translations with rare or out-of-vocabulary words.

- Scalability and Resource Consumption
  Training massive language models like these requires significant computational resources and electricity. The environmental impact of such resource consumption should be considered. There was no mention of such considerations.

- Data Bias
  The language model is trained on large amounts of data without any mention of any preprocessing done on it. This can lead to biased outputs from the model.

4.5 Suggestions

Below is a list of suggestions that we had for the authors regarding the limitations discussed earlier.

- Held-out Perplexity on Specific Genre Data
  As perplexity is sensitive to genre mismatch, a held-out test set that reflects the target genres (news, broadcasts) could perhaps be used to provide a more accurate perplexity measure.

- Acknowledge the Limitations of the BLEU Score
  Acknowledge the limitations of the BLEU score and consider including human evaluation for a more nuanced understanding of translation quality.

- Data Bias and Potential Mitigation
  Discuss the issue of data bias and potential mitigation strategies.

- Mitigation of Power Consumption
  Although the power consumption of training these models is not enormous, it is still a good practice to consider mitigations for future cases.

5 Discussion on Ethical Consideration

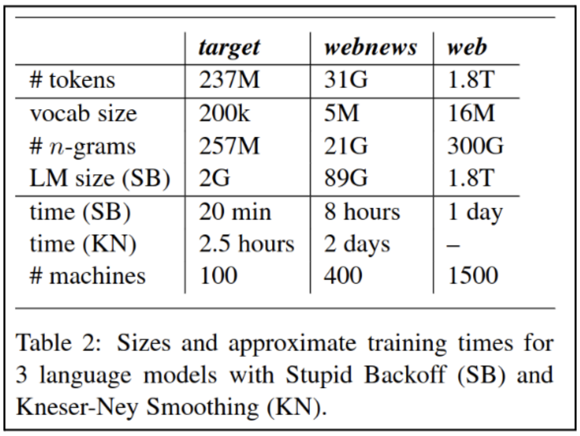
- **Energy Consumption**
  In recent years, a major concern that has become prominent in the discourse surrounding Large Language Models is the energy it takes to train these models. Indeed, the model presented by Brants et al. is no different because it took a significant amount of computational power to train these models. In particular, the model trained on the largest dataset web took one day on fifteen hundred machines. Figure **??** provides further details.
  It is important to note that this training period is incredibly modest compared to modern transformer-based LLMs. However, the researchers fail to include relevant power usage statistics, and we are left to speculate on the energy costs associated with this model.

- **Data Usage and Privacy**
  The datasets in use consist mainly of public-domain information. However, the largest dataset (web) consists of data gathered across the web. While most of this data is probably public domain, the authors fail to do their due diligence in ensuring the information can be used to train a model that may be used in a for-profit scenario. As such, there are legal copyright concerns.
  Furthermore, many users on the web may not consent to having their data used to create a product. The ongoing discourse regarding

user data usage for targeted ads and analytics has corroborated this idea. This consent is especially relevant should the resulting LLM be used for profit. As such, the web dataset needed to be constructed more carefully, and if it were carefully built, the process needed to be described.

- **Bias and Fairness of Training Data Sets**
  The issue of bias and fairness in language models trained on large datasets is a significant concern. These models are prone to developing biases from their training sets. This can lead to discriminatory or unfair outcomes, particularly in translation tasks. The paper mentions that the training involved four distinct datasets, varying substantially in size, ranging from 237 million to 2 trillion tokens. A notable risk here is the dominant influence of biases from the larger datasets, potentially overshadowing those in the smaller ones. Additionally, the composition of these datasets is crucial. With half of them being news-based, there is a risk of bias towards news terminology and perspectives. This skew could affect the model's performance in translation tasks, especially in translation contexts that are not news-related.
  Notably absent in the paper is a discussion on the ethical implications of such biases. Addressing potential biases in the training data and mitigating their impact on the model's output is crucial for ethical AI development, such as diversifying data sources or implementing fairness-aware algorithms.

- **Potential impacts of misuse of LMs**
  While the paper thoroughly focuses on the technical information, it does not shed light on the potential misuses of translation models, which can extend to spreading false information, privacy breaches and limiting freedom of speech.
  If used maliciously, advanced translation tools could enable the spread of misinformation across language barriers more effectively, impacting public understanding and discourse.Machine translation could be employed to alter the tone or content of messages or news, potentially distorting the original meaning to suit specific agendas. Hence the authors should also shed some light on the free and fair usage of such models and incorporate moderation to mitigate potential misuse of such generation models.



	target	***webnews***	***web***
# tokens	237M	31G	1.8T
vocab size	200k	5M	16M
# n-grams	257M	21G	300G
LM size (SB)	2G	89G	1.8T
time (SB)	20 min	8 hours	1 day
time (KN)	2.5 hours	2 days	–
# machines	100	400	1500

Table 2: Sizes and approximate training times for 3 language models with Stupid Backoff (SB) and Kneser-Ney Smoothing (KN).

Figure 1: Training times for the language models

6 Ranking on Various Attributes

This section ranks the papers on certain attributes. These rankings are done to promote a healthy discussion and do not aim to detract from the effort put in by the authors.

6.1 Soundness - Score 3

The central idea of the paper was a language model that allows scaling to very large amounts of training data, and the authors provided a thorough methodology and experimentation while exploring this idea.
The only reason we did not give it a four was because the experiment's results, although extensive, required a certain level of knowledge in order to comprehend the metrics themselves. Since the authors did not clearly convey the meaning of these metrics, we detracted a perfect score.

6.2 Presentation - Score 3

The use of graphs and tables could have been extended to elaborate further on details. Figure 1, for example, defines how vocabulary is generated in a distributed framework using mapReduce, but its internal workings are slightly ambiguous. Furthermore, effort could've been put into explaining the equations in a more elaborate manner (Equation 1 could have been described in more detail to set the stage for language modelling).
However, we commend the authors for communi-

cating their ideas concisely without drowing the reader in domain specific jargon.

6.3 Contribution - Score 4

Considering the fact that this paper introduces a new smoothing technique that has become widely used today, the paper should get a 4. On top of this, introducing an efficient distributed training system that could train language models on billions of tokens in the pre-neural network language model era is also a significant achievement.

6.4 Overall Rating - Score 8

Overall, we would rate this paper an 8. The paper provides a good insight into the state of language model research at the time of publication. It also explored new ideas and improved on already existing ones to deliver a valuable contribution to the domain of statistical language modelling by using larger, distributed models.

References

Thorsten Brants, Ashok C Popat, Peng Xu, Franz J Och, and Jeffrey Dean. 2007. Large language models in machine translation.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Hermann Ney and Stefan Ortmanns. 1999. Dynamic programming search for continuous speech recognition. *IEEE Signal Processing Magazine*, 16(5):64–83.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.