REVIEW OF DATA COMPRESSION TECHNIQUES

A PREPRINT

Aditya Meharia
School of Computer Engineering
Kalinga Institute of Industrial Technology
Bhubaneswar, India 751024
`adityameharia14@gmail.com`

Junaid H Rahim
School of Computer Engineering
Kalinga Institute of Industrial Technology
Bhubaneswar, India 751024
`junaidrahim5a@gmail.com`

March 4, 2020

ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Keywords Machine Learning · Machine Translation · Decoding · Improvement

1 Introduction

Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) have been the cutting edge solutions in the field of Machine Translation. The NLP community has shown keen interest in SMT due to its ability to perform more accurate translations with less parallel corpora. Phrase based SMT systems have been the best in terms of performance as they use a phrase based model to overcome the disadvantages of a word based model. These models work by translating sequences of words of varying length rather than translating word by word. A phrase based SMT

system usually consists of three models viz. the language model, the translation model and the distortion model. Mathematically, these models are represented as:

$$e_{best} = argmax_e P(e|f) = argmax_e [P(f|e).P_{LM}(e)]$$

Here, f and e mean input in foreign language and output in english respectively. e_{best} is the translation e with the highest probability, given the foreign piece of text f. $P_{LM}(e)$ is the language model, it assigns a probability of correctness in the particular language to e. $P(f|e)$ is the translation model which is expressed as:

$$P(f^I|e^I) = \prod_{i=1}^{I} \phi(f^i|e^i) d(start_i - end_{i-1} - 1)$$

where, $\phi(f^i|e^i)$ is the phrase translation probability, it is the probability of f^i being translated to e^i, it is learned from the parallel corpus. $d(start_i - end_{i-1} - 1)$ is the distortion probability. It is the distance between words in source and translated text, settled with an exponential cost.

Decoding is a generate and score process in which the best translation is searched in the space of all the possible translations. The decoding problem for statistical machine translation is NP-complete. Thus, heuristic search methods are used to find the best translation. Traditional decoding methods make use of the Beam Search algorithm to search for the ideal translation in a huge number of hypotheses. Naturally the poor hypotheses need to be pruned out. The standard implementations of decoding use threshold pruning and histogram pruning algorithms to achieve this. Threshold pruning removes the hypotheses from the stack that have a score lower than a fixed threshold value (beam threshold), in histogram pruning, only a fixed number of hypotheses are kept in the stack (stack size).

Beam threshold and Stack size play an important role in translation accuracy and decoding time. In most of the standard implementations, the values of these parameters are already fixed, there has been a study where the parameters are dynamically selected according to the source and target languages using a Machine Learning based approach. The decoding time and translation accuracy have shown considerable improvement when the parameters are dynamically predicted and set by the machine learning model. The machine learning based approach outperforms a NMT system and a SMT system with fixed values for stack size and beam threshold. Our work aims to test this machine learing based approach with more experimental rigour and find reasons for this improvement.

1.1 Previous Work

Many improvements has been made to (Moses) the open source toolkit for statistical machine translation, its output quality and speed has been improved when compared to the previous stack based decoding approach using new decoders [] [] and also many improvements were made in language models which are very fast and efficient [] []. The addition of phrase table which can be loaded whenever required by the SMT decoder [] has shown a great reduction in memory requirement and initial loading time, which was later further improved by compressing the on-disk phrase table and lexicalized re-ordering model [].

Great pruning approaches like cube-pruning and cube-growing algorithm in [] gives us a single parameter which allows us to control the tradeoff between translation accuracy and speed.The Moses decoder was introduced by Koehn et al. [], where the traditional SMT uses beam search for finding the best translation from the subset of hypothesis (possible translations) stored in a stack. As different inputs will require different parameters which were somehow related to the input sentences. In SMT the same decoder parameters were applied, no matter what the source text and is not possible for the user to decide the set of parameter's values according different input texts, there was no feature for this task. It is known that there is a tradeoff between translation accuracy and decoding time in machine translation. There were efforts made to reduce the decoding time or increase the translation accuracy but all approaches were for the training model (MERT) which would tune its parameters [] or in improving the decoding algorithm []. Until in D. Banik et al. [] where they proposed a machine learning based parameter selection technique to achieve better decoding, where they have shown that parameter's values are a very crucial factor for decoding time and translation accuracy. They trained a classifier where it will take the source text as the input and classify them into two parameter classes i.e stack size and beam threshold based on the input text, here bigger stack size means better translation accuracy but poor decoding time and a bigger beam threshold will give fast results but in the cost of less good translations, so the classifier will help to select the best parameter class for a given input text.

The features they took for the classification task were directly related to the complexity and structure of the input text, which are percentage of comma (,) in the text, percentage of long sentences in the text, average words per line in the

text, percentage of stop words. They have used the CN2 unordered algorithm [], [] as the classifier model, whose central concepts are based on Algorithm quasi-optimal learning (AQ algorithm) [] and the Iterative Dichotomiser 3 (ID3) algorithm. It was seen that the classification task could be made better with properly handling the low data constraint and with better feature selection, feature engineering and using a different algorithm approach for the model, all of which we propose in this paper.

1.2 Our Approach

This space is to explain our approach

- motivation - How the classification task can be made better with testing out different models with better features and hyperparameter tuning - methods - Our feature selection and engineering techniques - tools - results in short

1.2.1 Headings: third level

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Paragraph Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

2 Examples of citations, figures, tables, references

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

[1, 2] and see [3].

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

> Hasselmo, et al. (1995) investigated…

`https://www.ctan.org/pkg/booktabs`

2.1 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

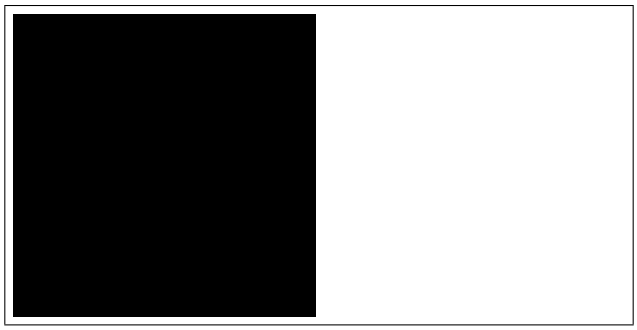

Figure 1: Sample figure caption.

Table 1: Sample table title

	Part	
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

See Figure 1. Here is how you add footnotes. [1] Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

2.2 Tables

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

See awesome Table 1.

2.3 Lists

- Lorem ipsum dolor sit amet
- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

References

[1] George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014.

[2] George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014.

[3] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.

[1] Sample of the first footnote.