Large Language Model Optimization

by Paulius Kančauskas

Problem Overview

Large language models are generating text one step at the time. They are taking sequence of tokens as an input and predicts which next value is most probable. That means that for every step in text generation you have to input initial sequence + already predicted tokens over and over for every next token to predict (Figure 1.0).

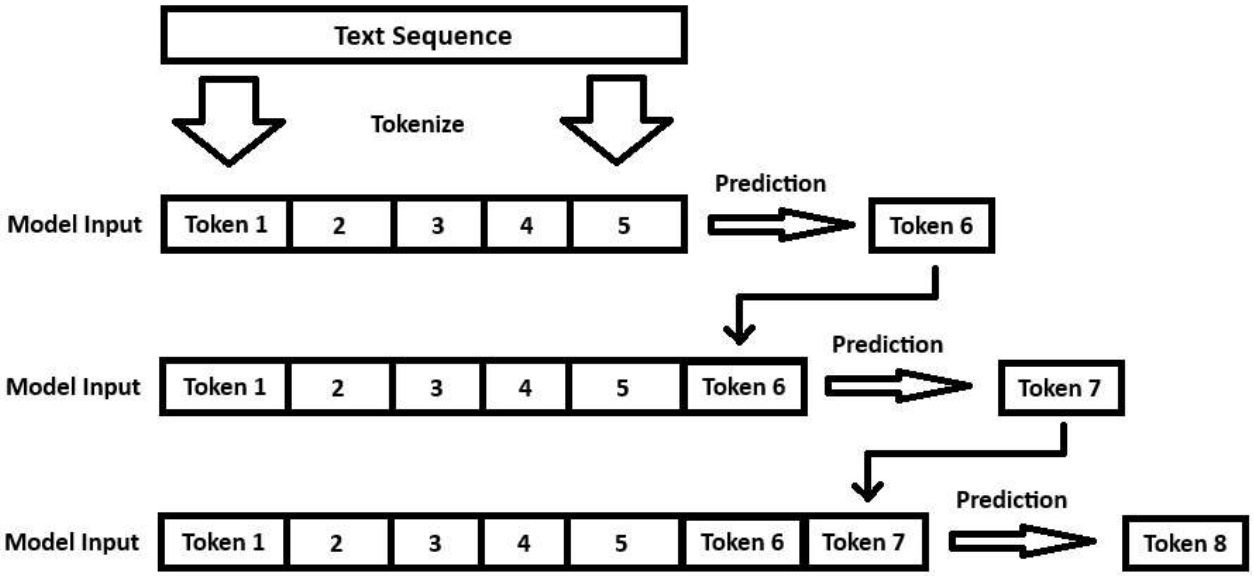

Figure 1.0 Example of sequence generation

There is a lot of repetitive work. We can calculate total tokens processed using this formula:

$$Total\,tokens\,processed = (t + \frac{g-1}{2}) \times g \quad (1)$$

t – initial token amount.
g – generated token amount.

For example, if we have 200 initial tokens and have generated 30 tokens, we have processed 6435 tokens in total. 200 tokens are approximately 130 words. And 30 tokens are about 20 words. So, it is quite a lot for such a small text. We can see from the graph below that processing has a similar to quadratic increase. (Figure 2.0). Imagine how many tokens would be processed in text analysis and summarization or article generation where might be thousands of tokens analysed and generated.

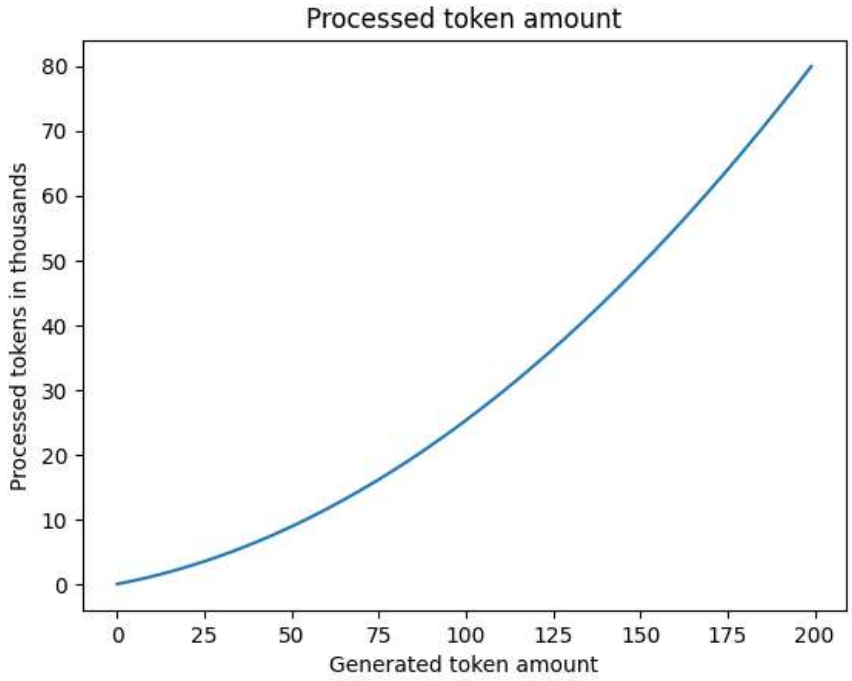

Figure 2.0 Processed tokens in thousands depending on generated token amount

Lets see how it affects token generation speed. In Figure 3.0 we can see that till the 100 token sequence length output, the average time is actually getting lower. It might be because of software and hardware overhead. Over 100 the average time per token is increasing.

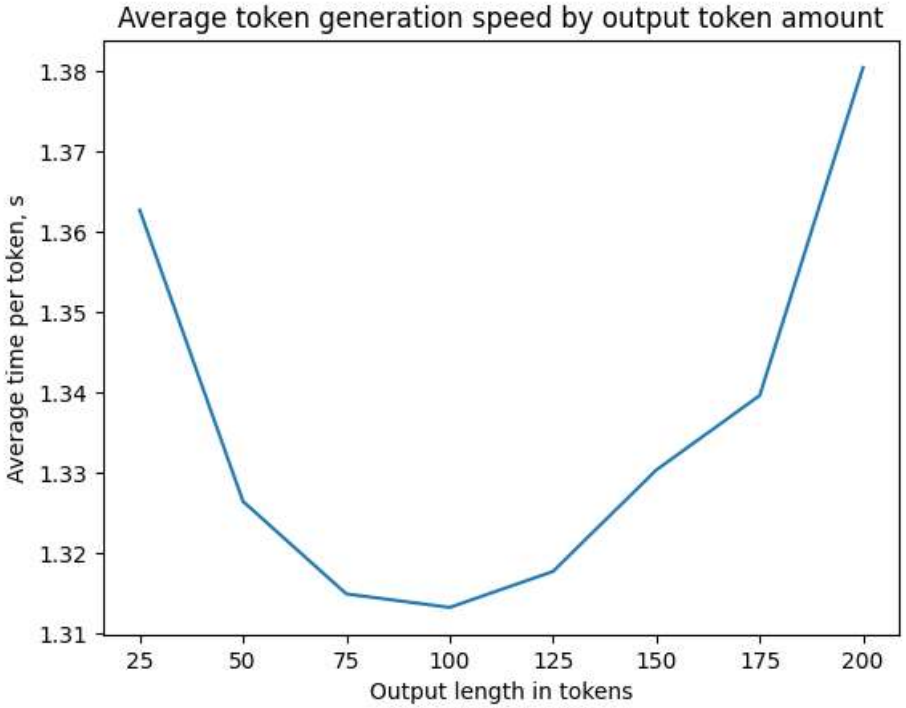

Figure 3.0 Average token generation speed depending on output sequence length.

Even though difference looks not that big, but it might become very visible if you have hundreds of thousands of request per pay. It would add up to your monthly and yearly total usage of

hardware and time. Even if we have difference of 2 seconds it adds up to ~83 hours per month of hardware usage if you have 5000 requests per day. Of course, everything comes down to hardware, electricity, maintenance and services cost as well as the user experience.

There are some solutions that can improve the performance. I will review some of the general optimization techniques to see how much we can improve the performance of the model.

Optimization techniques

Number Precision

Popular frameworks like Pytorch or Tensorflow can be easily set up to use different precision for model inference and training. I will use float16 and for comparison bfloat16. Even though they are both 16 bit precisions they have a little different structure. Bfloat16 has more dedicated bits to exponent part and less for fraction compared to float16. Exponent part is actually the same as float32, just fraction part is cut off. Models usually by default are 32 bit precision, loading model to 16 bit should not only increase performance but also reduce the size of the model.

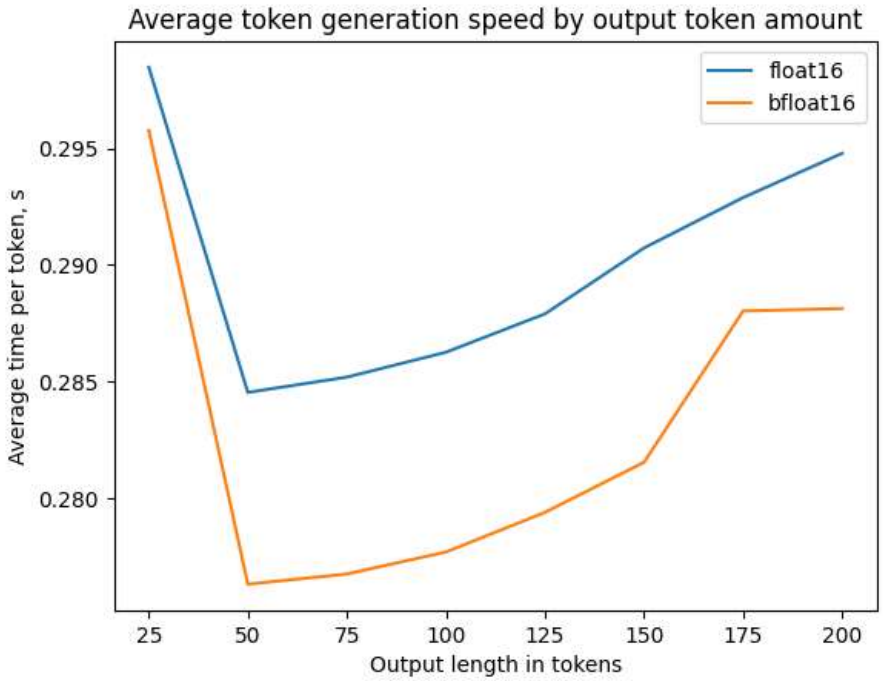

Figure 4.0 FLoat16 and bfloat 16 precision average token generation time.

Bfloat16 precision is slightly faster than float16 (Figure 4.0) and way more faster than float32 comparing to Figure 3.0. It seems that it is more than 4 times faster. Memory usage is also lower (Figure 5.0). About two times as expected from two times lower precision.

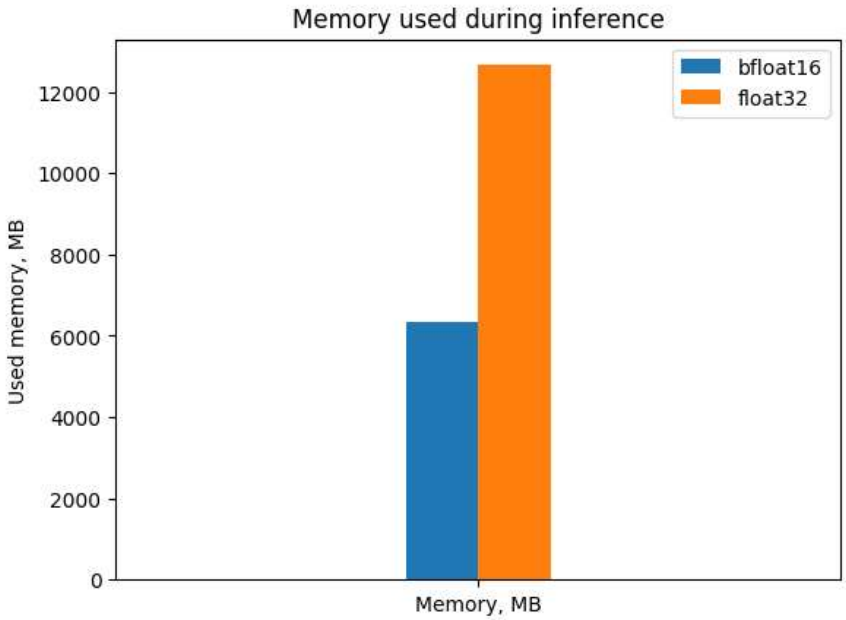

Figure 5.0 Memory usage of float32 and bfloat16 precision.

Even though we see great improvements, it should be considered that lower precision might not be suitable for some applications, like scientific, financial, regression solutions, where actual number values are important or small changes in parameters have big impact on model outcome. Also, you need the hardware that supports it, for example Nvidia GPUs with tensorcores, built in mind for such tasks.

KV Caching

Other useful optimization technique is the Key-Value caching. As the transformer models with attention mechanisms are doing calculations in parallel (like matrix multiplications) some of the values can be pre-calculated once and used later. This helps with repetitive calculations for each new token.

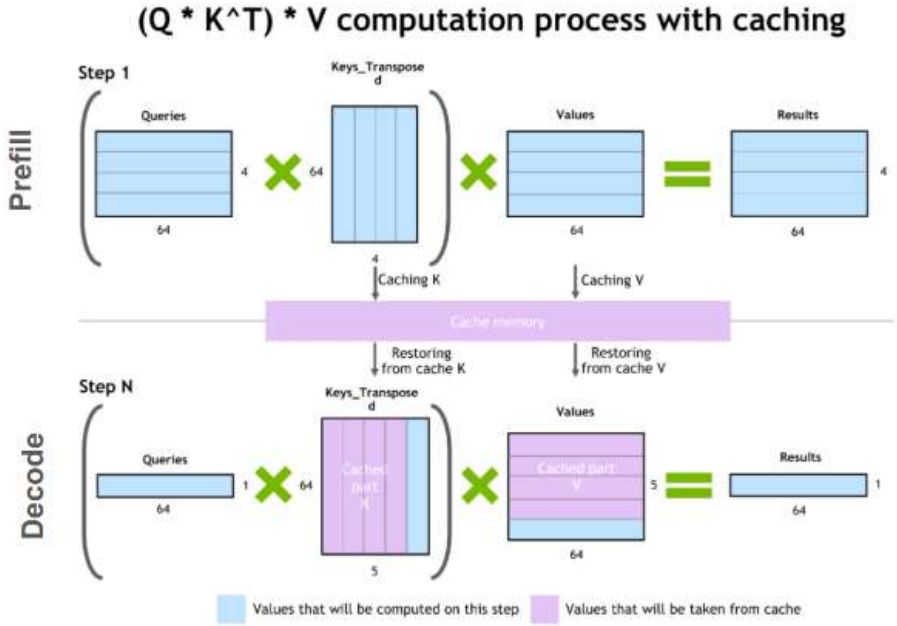

Figure 5.0 Caching mechanism scheme.
Source https://developer.nvidia.com/blog/mastering-llm-techniques-inference-optimization/

Caching increased our performance almost 10 times (Figure 6.0). Note, that this time I generated up to 1000 tokens. I wanted to see if average token time starts to increase as we saw before, but it seems that it is even getting lower, at least up to 1000 tokens.

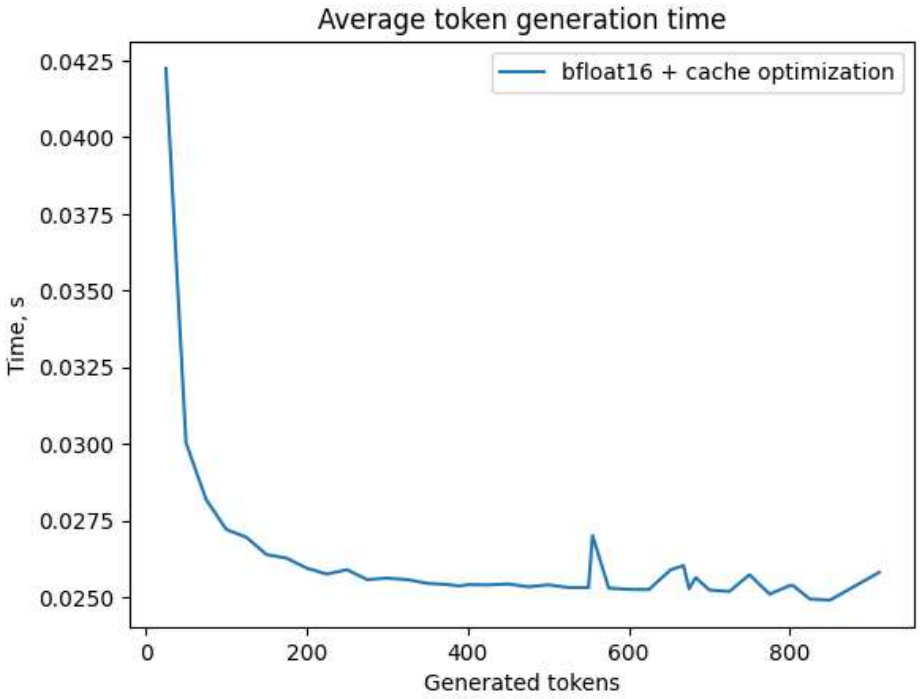

Figure 6.0 Average token generation time using bfloat16 precision and caching optimization.

Cache takes extra space for prefilled values, so lets see how memory usage is doing (Figure 7.0).

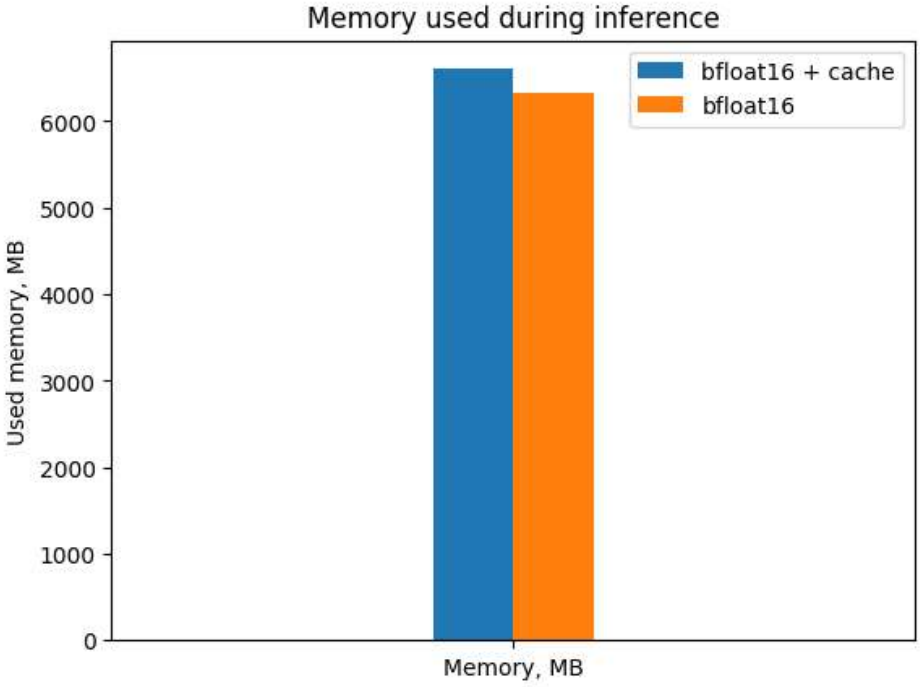

Figure 7.0 Memory usage using bfloat16 with and without cache.

This doesn’t seem too bad but you have to keep in mind that attention models scale quadratically when input sequence increases and also be aware that during batching every sequence in batch will have their own cache so, usage may increase in no time.

Before we move on let’s take a quick look at total performance gains.

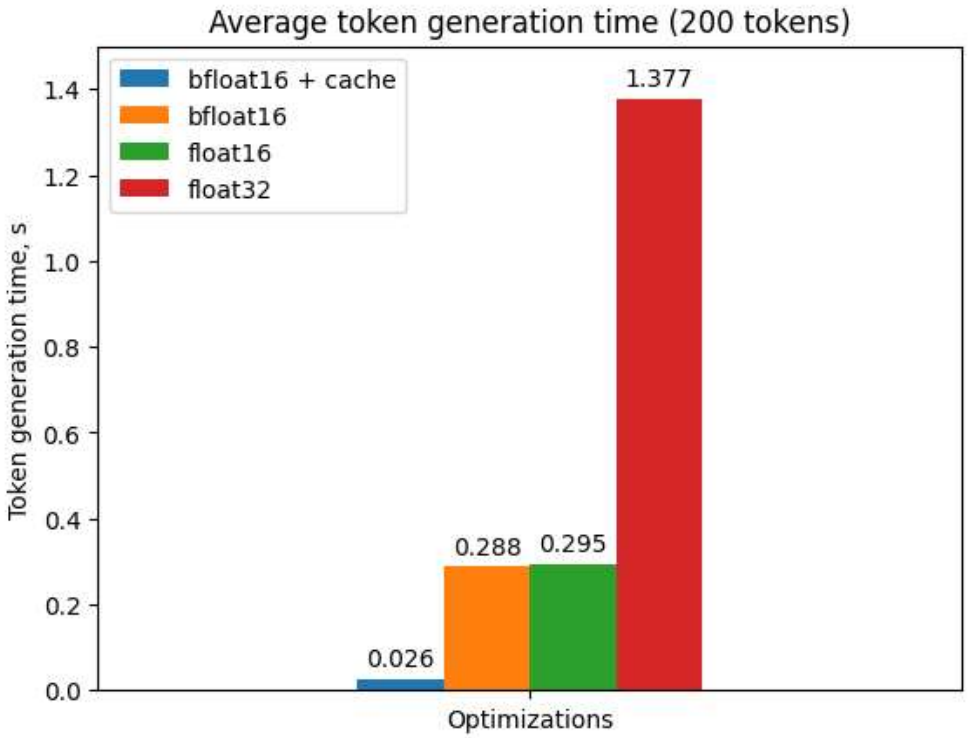

Figure 8.0 Average token generation time when generating 200 tokens.

By using these techniques we managed to improve our performance by over 50 times. Most frameworks have already adopted these optimizations, so it is not difficult to implement them. In some frameworks they might be applied by default and if not, it is worth taking a look if you can enable them.

Algorithmic optimization

This time we will look at the actual question answering, how much time it takes to read Labour Code and to generate the answer. Most simple way is just to concatenate Labour code together with the question and insert to the model.

For this task I will be using Llama 3.2 3B Instruct model. This model has 3 billion prameters and is pretrained by Meta.

Firstly, we should get our base performance line for comparison. So, I will input the Labour Code and ask several questions. For visualization purposes questions will be numbered as cases.

Table 1.0 User input numbers

Case	User Input
1	What type of job contracts there are?
2	I am working in shifts. Sometimes I have to work at night. Do they need to pay more for working at night?
3	What are legal reasons to quit a job?
4	I am planning to become a mom. What should I know about my rights at the job?

Lets take a look at total answer generation time (Figure 9.0). Case 3 took longest, almost 110 seconds and the quickest was case 2, about 50. You will find generated answers in Appendix 1.

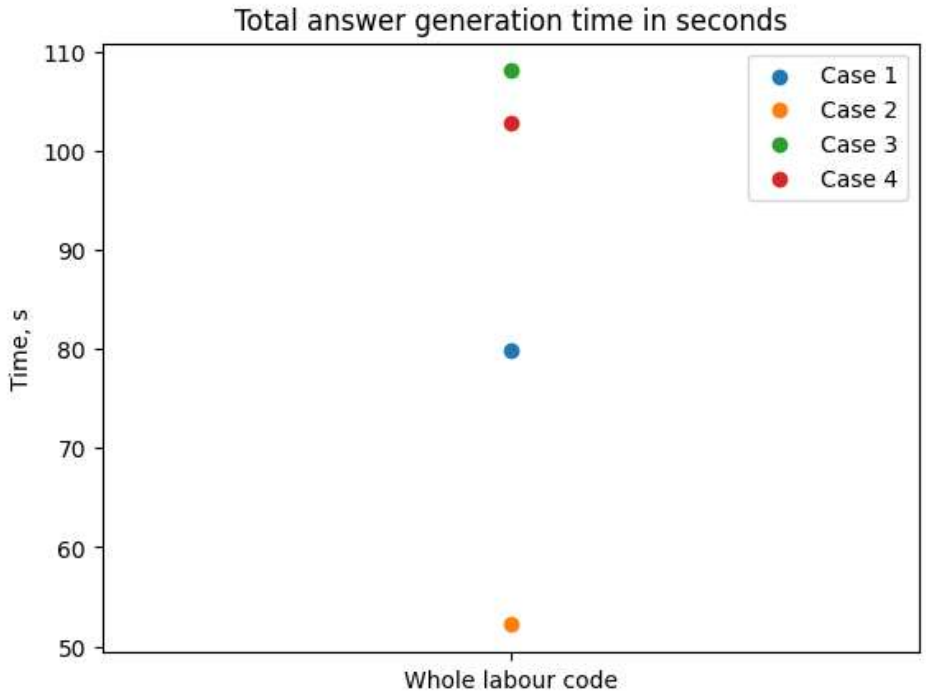

Figure 9.0 Total answer generation time. Input is whole labour code with user question.

Using whole labour code as an input might not be very efficient way. Especially, when we have simple, short questions. It would be more efficient to give only related articles, those who might be useful to user situation. But then we need to filter them. For filtering I am using the same model. I input the article one by one and ask a question if this article is related. Then I concatenate filtered articles and feed to the model to answer the user question. At first it might look that I still feed the whole labour code just article by article and there would not be much improvement. But if we look at the Figure 10.0 below we can see that we have very good improvements.

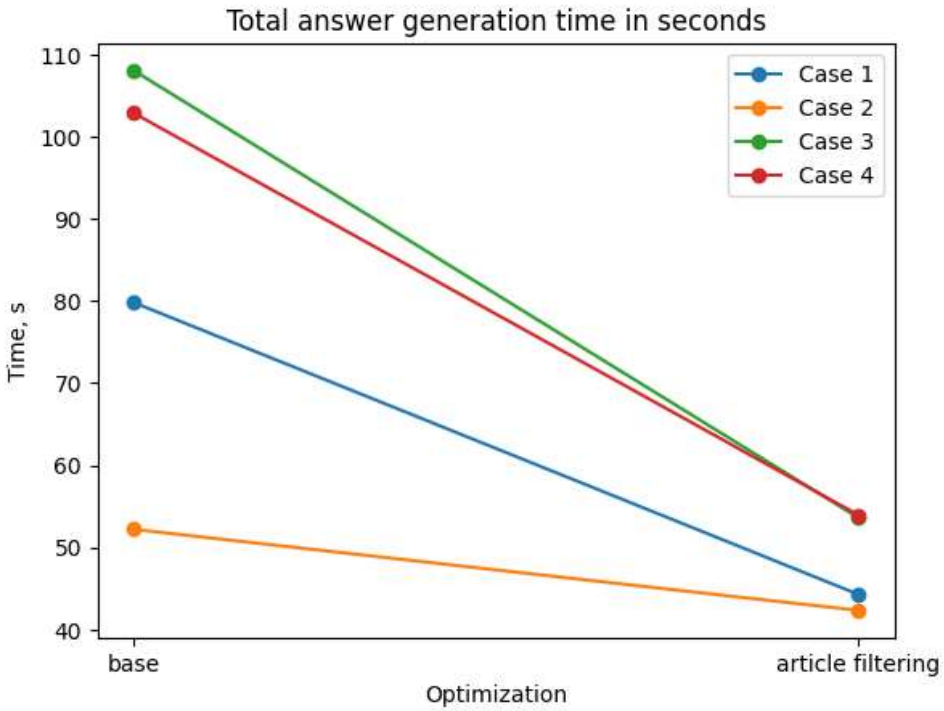

Figure 10.0 Total answer generation time using whole labour code and separate articles.

The intuition behind this method is that you need to generate only several tokens to tell if article is related or not, so it doesn't take too long to filter them. And depending on the question you might select only 10 or 5 articles, what is way less than whole labour code with more than 250 articles. So, when generating the answer that might be quite long you are processing way less tokens that are not relevant to question.

The takeaway of the filtering is that it takes processing time before generating the answer. To reduce it we can input articles in batches to take advantage of parallelism. In the Figure 11.0 we can see how much time filtering takes depending on the batch size.

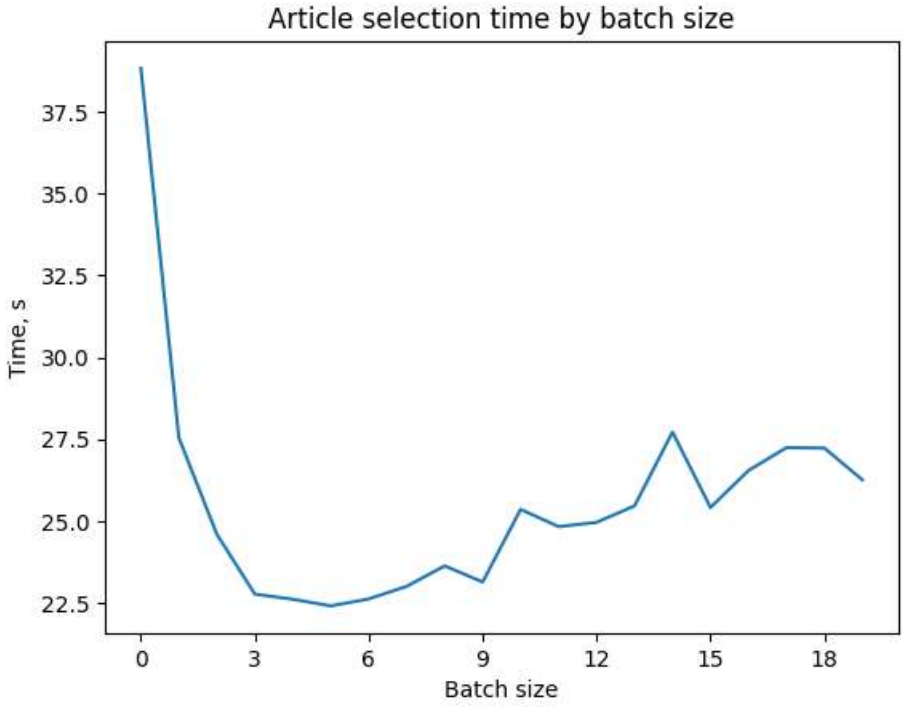

Figure 11.0. Article filtering time by article amount in batches.

Article filtering time is improving until batch size of 5-6 articles. After that putting more articles to batch starts to increase processing time. It is because articles are in different sizes. Some articles are long, taking a couple of pages, some are shorter and has just several sentences.

Why it is a problem? Sequences in the batch has to be the same size, because it is treated like matrix during calculations. To achieve that we need to pad shorter sequences to match the length with longer ones. So, if there is one article that is very long, the bigger the batch is, the more short articles can fall to the same batch with it. To avoid this we can try to optimize the batching process, so that short articles are combined with other short articles and long with longer ones. In this case, batch size is determined by total tokens of batched articles. Batch size is bigger if articles are short, and smaller if articles are long. In the Figure 12.0 below we can see how processing time depends on batch size measured by tokens.

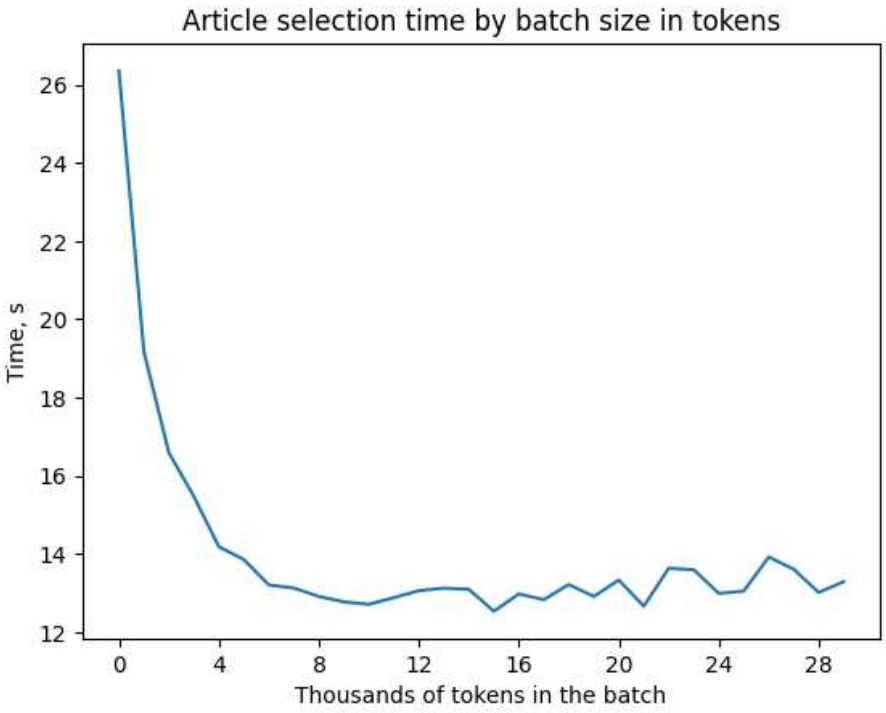

Figure 12.0. Article filtering time using optimized batches.

The processing time is reduced by half and optimal batch size is about 10000 tokens. It might ba a batch of 5 articles or it might be a batch of 20 articles, depending on article length. Optimization is achieved by tokenizing every article, sorting them by token amount and grouping them to have 10000 tokens in group. In the Figure 13.0 below we can see how total times have improved.

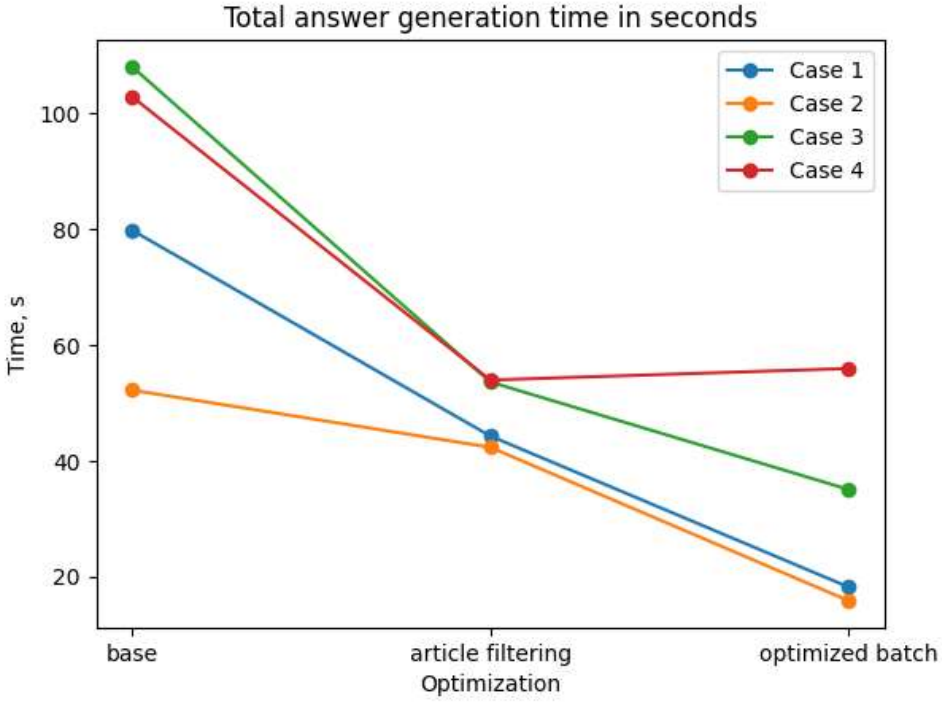

Figure 13.0. Total answer generation time by optimization method.

Case 4 took longer than before, but we should be aware that answers are not identical, so total token amount in answers differ. In the Figure 14.0. below we can see how average token generation time has improved.

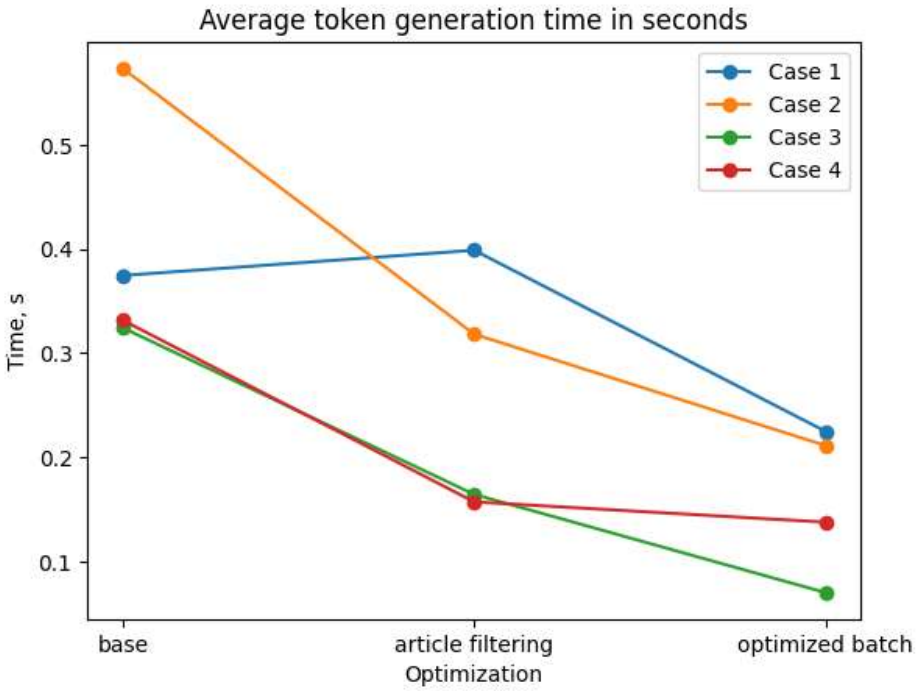

Figure 14.0. Average token generation time by optimization method.

In total, answer generation time was improved about two times on average.

Conclusion

There are various methods to optimize your model and results are quit good. The inference time is reduced 50 times using lower precision and caching. Also, reduced by half the processing time using smart approach to article filtering and batching, what in total adds up to 2 x 50 = 100 reduction in processing time. Of course, results may vary depending on the model, hardware and the use case, but there is much potential and most optimization methods are included in popular frameworks, so it is not difficult to implement them.

Also, there are other optimization possibilities that are not mentioned here and worth looking into, like quantization to 8 bit or even 4 bit, model distillation or pruning that might help not even to speed up the inference but also to reduce the model size.

Appendix I

Case 1. *The user is asking about the different types of job contracts.*
Filtered articles (11)

Here are the types of employment contracts mentioned in the provided articles:

1. **Article 66. Types of Employment Contracts**
   - Open-ended employment contract
   - Fixed-term employment contract
   - Temporary agency employment contract
   - Apprenticeship employment contract
   - Project-based employment contract
   - Job share employment contract
   - Multiple-employer employment contract
   - Seasonal employment contract

Case 2. *The person is asking if they should be paid more for working at night shifts.*
Filtered articles (3)

Based on your question, the relevant articles are:

* Article 117: The Specifics of Working-Time Arrangements for Working at Night
* Article 144: Payment for Work on Days Off and Holidays and Overtime Work, and Compensation for Employees Whose Work is of a Mobile Nature or Involves Trips or Travelling

Case 3. *The user is asking about the legal reasons for terminating a job.*
Filtered articled (18)

Here are the articles related to the legal reasons to quit a job:

1. **Article 56. Termination of an Employment Contract on the Initiative of the Employee for Valid Reasons**
   - The employee has been on idle time without any fault on the part of the employee for more than 30 consecutive days, or for more than 45 days over the past 12 months;
   - The employee has not been paid the full remuneration (monthly salary) due for two or more consecutive months, or the employer has failed to fulfil, for more than two consecutive months, the obligations established by the labour law provisions regulating safety and health at work;
   - The employee is unable to properly perform his or her job function due to an illness or disability, or due to the fact that he or she is caring for a family member (child/adopted child, father/adoptive father, mother/adoptive mother, husband or wife) at home for whom a special need for permanent nursing or permanent care/assistance has been established in accordance with the procedure established by legal acts;

- An employee working under an open-ended employment contract has reached the statutory age of old-age pension and has acquired the right to full old-age pension while working for that employer.

2. **Article 57. Termination of an Employment Contract on the Initiative of the Employer without any Fault on the Part of the Employee**
- The job function performed by the employee has become superfluous due to changes in work organisation or other reasons related to the employer's activities;
- The employee is not achieving the agreed performance outcome according to the performance improvement plan provided for in paragraph 5 of this Article;
- The employee refuses to work under changed indispensable or supplementary employment contract terms or to change the type of working-time arrangements or place of work;
- The employee does not agree to continuity of employment relations in the case that the business or part thereof is transferred;
- A court or body of the employer has taken a decision ending the employer.

Case 4. *The user is seeking information about their job rights as a prospective mother, specifically what they should know about their rights while planning to become a mother.*
Filtered articles (12)

Based on your question, I have selected the following articles that are relevant to your situation:

1. **Article 52: Remote Work** - This article does not directly relate to your question, but it mentions that remote work may be assigned at the request of the employee or by agreement of the parties.
2. **Article 61: Restrictions on the Termination of an Employment Contract** - This article does not directly relate to your question, but it mentions that an employment contract with a pregnant employee during her pregnancy and until the baby reaches four months of age may be terminated by mutual agreement, at her initiative, at her initiative during the trial period, in the absence of the will of the parties to the contract, or when a fixed-term employment contract expires.
3. **Article 128: Granting Annual Leave** - This article does not directly relate to your question, but it mentions that pregnant employees are entitled to 70 calendar days before childbirth and 56 calendar days after childbirth (or 70 calendar days in cases of complicated childbirth or when more than one child is born).
4. **Article 132: Unpaid Leave and Unpaid Time Off** - This article does not directly relate to your question, but it mentions that unpaid leave of up to 14 calendar days may be granted to an employee raising a child under the age of 14.
5. **Article 134: Child Care Leave** - This article directly relates to your question, as it mentions that child care leave may be granted until the child reaches three years of age, and that employees entitled to this leave may take it in turns.
6. **Article 131: Special Leave** - This article directly relates to your question, as it mentions that special leave includes pregnancy and childbirth leave, paternity leave, child care leave, educational leave, sabbatical leave, and unpaid leave.
7. **Article 132: Pregnancy and Childbirth Leave** - This article directly relates to your question, as it mentions that eligible employees are entitled to 70 calendar days before childbirth and 56 calendar days after childbirth (or 70 calendar days in cases of complicated childbirth or when more than one child is born).

8. **Article 137: Unpaid Leave and Unpaid Time Off** - This article does not directly relate to your question, but it mentions that unpaid leave of up to 14 calendar days may be granted to an employee raising a child under the age of 14.