NLP Course AI Talent Hub

Nikita Venediktov

May 2023

Abstract

The use of Supervised Finetuning (SFT) allows us to greatly improve the model before the Reinforcement Learning from Human Feedback (RLHF) stage. The created synthetic dataset of job summation, allows adapting a large language model (LLM) to the job search tasks. Based on the results of the work the dataset uses extractive and abstractive summarization, as well as using expert labeling. Link to my project code right here: `https://github.com/NikitaVenediktov/Dataset_for_SFT/tree/main/summarization`.

1 Introduction

The work is aimed at reducing the problem of using LLMs to solve specific problems. The specifics of the problem considered in this paper is that a job site requires a job summarization service. This one will be able to display a preview of the job on the homepage. The main limit is no more than 200 symbols. The service will offer the employer a choice of several generation results, and the user will choose the best one or write his own. The business process is set up so that we save the user's choice and then use it for RLHF. At first we tried to bypass the SFT stage and refine the model for our tasks at the RLHF stage, but the basic generation was low. There are many reasons for this, chief among them:

- Pre-learning on articles and news;
- Hallucinations.

Since such a dataset does not exist, it became necessary to create a dataset for additional preliminary training on the tasks of the HR segment.The uniqueness of the dataset is solved in that not only a manual labels is found in itself, but also checked labels using extractive and abstract methods.

1.1 Team

Project for Rabota.ru

Nikita Venediktov - development of a dataset, implementation of a pipeline for data preprocessing and summary generation;

Mark Panenko - idea, raw data, GPUs.

2 Related Work

In this section, you will describe in details the existing approaches to the problem you work on. For each approach, you need to provide a reference.

The article fully describes the learning curve of the model and the importance of SFT [Huyen, 2023]

The first high-level introduction to summarization methods I got from the article 'Text summarization techniques' [Shariati, 2021]

It was interesting to see a competitor's solution based on a different neural network architecture [Habr/HeadHunter, 2022]

Fresh master's thesis on Summarization and keyword extraction on customer feedback data [Skoghäll and Öhman, 2022]

To write the extractive pipelines, I used an article from jet brains [Academy, 2022] and sumy library documentation [Belica, 2018]

3 Model Description

In general, the task of the model looks like seq2seq or text2text.

Let me describe the service model that will use my dataset.

1. Transformer selection (mBART, T5, GPT)
2. Synthesizing dataset for adaptation to our needs
3. Supervised finetuning (SFT) stage
4. RLHF - using the model from the 3rd stage we generate and hand-sketch dataset for the first time
5. In production and later use user feedback to generate a large training dataset
6. Qualitative model of vacancy summation

Full training pipeline is presented on Fig. 1.

4 Dataset

On the Tab. 11 you can see that there were almost 50k jobs in the original dataset, and 9k of them had an extra skill description. Features of vacancies:

- problem in extracting the essence - company stories;
- inability to rank offers in the job - one line;
- the problem is in the accuracy of extraction - if it is good, but a lot;
- we should have adapted pre-processing - a lot of numbers slicing bulleted lists into sentences.

The prefetching included job-specific processing with regular expressions.

Was the choice of what to extract words, phrases, sentences? I chose sentences because they have more words and less chance of getting incoherent text.

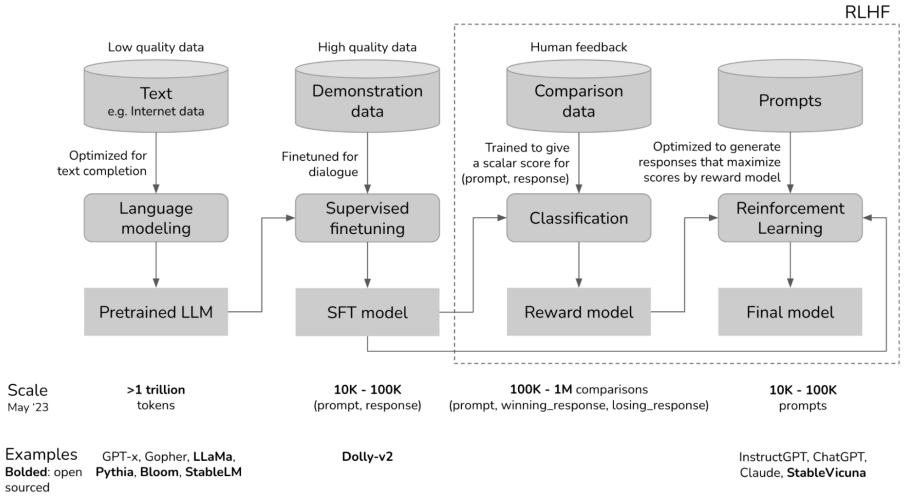

Figure 1: Full training pipeline

column	amount
description	49987
skills	8822

Table 1: Dataset from database

5 Experiments

5.1 Metrics

Summarization metrics such as BLUE or ROUGE, other things being equal, do not allow us to evaluate the effectiveness of summarization for adequacy, especially abstract In paper [Skoghäll and Öhman, 2022]. The summaries from the model based on LSA theory are very hard to follow and often did not make any sense in regards to capturing the context and having the worst Rouge metrics. But for our tasks LSA gives one of the best results.

According to human evaluation, the model based on Textrank and LSA are the best performing models.

5.2 Experiment Setup

I used sumy library for extractive summarization. It implements the main methods of extractive summing and is based on the NLTK library. Using NLTK it is easy to implement Russian tokenizer and stemmer. The full extractive pipeline looks like this:

1. Clearing the job text from stop words, hyphenation, and lists using regular

expressions

2. Cleaned text is processed by a streamer

3. Tokenization

4. Ranked and issued 3 sentences

If the length of the extractive summarization is more than 200 characters, then transfer the data to the extractive pipeline The full abstractive pipeline looks like this:

1. Lock the work of the generative model to 52 tokens, so as to get the generation of less than 200 symbols

2. Tune the generation parameters, so that there is not too free generation

3. Generation

The basic generation settings can be viewed in the github.

5.3 Baselines

To implement an extractive summarizer, we need to import the necessary libraries. There are several algorithms packaged together in the Sumy and NLTK for python. In the following, I will describe some of them.

LexRank Summarizer: This is an unsupervised approach inspired by Google's PageRank algorithm. It finds the relative importance of all words in a document and selects the sentences which contain most of those high-scoring words. The scoring of sentences is determined by using the graph matrix. This connectivity matrix is based on intra-sentence cosine similarity. The sentences are ranked according to their similarities.

TextRank Summarizer: This algorithm also gets inspiration from the page rank concept. It is more simplistic than LexRank. This platform removes sentences with highly duplicitous by using the post-preprocessing step.

Luhn Summarizer: It was published in 1958 by IBM researcher, Peter Luhn. It looks at the window-size of non-important words between words of high importance. Also, the sentences occurring near the beginning of a document gets higher weight. First, this algorithm determines which words are more significant towards the meaning of the document. Finding the most common words in the document and taking a subset of those that are not common words but still important are the next steps.

LSA(Latent Semantic Analysis) Summarizer: This brand-new algorithm combines term frequency with singular value decomposition. In python, this algorithm works as follows: — Convert a document into a vectorized bag of words using CountVectorizer library — Encode the original data into topic encoded data — Fit and transform single value decomposition on the bag of words using TruncatedSVD library — Determine the strength of each part of the sentence effectively — Extract out the final sentences from topics

KL(Kullback-Lieber) Summarizer: This greedy method adds sentences to the summary as long as the KL Divergence decreases, i.e, it focuses on the minimization of summary vocabulary by checking the divergence from the input vocabulary.

After a blind test conducted by experts were selected LSA and TextRank for further work. Next, using the rut5-base-paraphraser model, I rephrased the long extraction variants.

6 Results

The created dataset is sufficient for the SFT transformer for the task of summarizing vacancies. The output methods work very differently, here's an example from below:

Требования: Строительная компания приглашает на работу вахтовым методом. Вахта на Север 60/30. Заработная плата 150 тыс. рублей. Обратите внимание: опыт работы в аналогичной должности строго ОБЯЗАТЕЛЕН. Оформление по ТК РФ, з/п 2 раза в месяц на карту. Работодатель предоставляет проживание, 3-разовое питание, спецодежду, обувь.Компенсация проезда Информация о работодателе: Строительная компания.

Table 2: Input Vacancy.

Строительная компания приглашает на работу вахтовым методом. Заработная плата 150 тыс. рублей. Обратите внимание.

Table 3: Output sumy_LexRank.

Строительная компания приглашает на работу вахтовым методом. опыт работы в аналогичной должности строго ОБЯЗАТЕЛЕН.Строительная компания.

Table 4: sumy_Luhn.

Строительная компания приглашает на работу вахтовым методом. Вахта на Север 60/30. з/п 2 раза в месяц на карту.

Table 5: sumy_KLdiv.

Строительная компания приглашает на работу вахтовым методом. опыт работы в аналогичной должности строго ОБЯЗАТЕЛЕН. з/п 2 раза в месяц на карту.

Table 6: sumy_LSA.

Строительная компания приглашает на работу вахтовым методом. з/п 2 раза в месяц на карту. Работодатель предоставляет проживание.

Table 7: sumy_TextRank.

Also in this section, you could provide some results for your model inference. The samples could be found in Tab. 9.

And abstractive got good results, shown from below:

sumy_LSA с 2006 года специализируется на создании инженерных и слаботочных систем. контроль и координация работ по всем направлениям инженерно-технической эксплуатации объектов. организация и контроль своевременного выполнение графика ППР на системах здания. rut5_parapharaser 'С 2006 г. занимается созданием инженерных систем и слаботочных систем, контролем, координацией работ по всему направлению инженерной и технической работы объектов, организацией и контролем своевременного выполнения'

Table 8: sumy_LSA + rut5_parapharaser

sumy_TextRank В связи с активным развитием в нашу команду требуется профессионал в области эксплуатации крупных социальных объектов. контроль и координация работ по всем направлениям инженерно-технической эксплуатации объектов. необходимых для выполнения объемов работ и отчетность по фактически освоенным объемам и затратам. rut5_parapharaser В силу активного развития в нашей команде требуется профессиональный специалист в сфере эксплуатации крупного социального объекта, контроль, координация работы по всему направлению инженерной и технической работы объектов, необходим

Table 9: sumy_TextRank + rut5_parapharaser

column	before cleaning	after cleaning
summary_human	7000	7000
sumy_LSA	49000	10000
sumy_TextRank	49000	10000
rut5-base-paraphraser(LSA)	16000	10000
rut5-base-paraphraser(TextRank)	13000	10000

Table 10: Final dataset

7 Conclusion

In this paper, a new dataset for the vacancy summation problem has been presented, which has been developed to apply the Supervised Fine-Tuning method based on modern transformer models. The dataset contains a variety of vacancies from the rabota.ru website, and its goal is to provide high-quality and accurate summaries to help you find a job.

Pre-processing of the data, including noise removal, tokenization, and splitting into training, validation, and test sets, has been performed. A pre-trained transformer model was selected and prepared for pre-training on our dataset.

Preliminary results show the potential of the model after pre-training. It is expected that after pre-training on our dataset, the model will show significant improvement in vacancy summation, achieving more accurate and informative summaries.

The results of this work have practical relevance and can be applied in various areas related to job search and job text summarization. It can help job seekers get job information faster and more efficiently, as well as help employers in the process of selecting qualified candidates.

Future work on the dataset and model is planned to further improve their performance and the accuracy of job summarization. This includes extending the dataset, improving pre-processing algorithms, and retraining the model on more data.

The dataset synthesized in this way allows you to accelerate the process of pre-training of the model

type sum.	amount	symbols	time
extractive	20000	150-200	1 hour
abstractive	20000	150-200	1 day
labelers	7000	150-200	1 month

Table 11: Dataset from database

References

[Academy, 2022] Academy, J. (2022). Extractive text summarization. https://hyperskill.org/learn/step/25440.

[Belica, 2018] Belica, M. (2018). Python library sumy. https://github.com/miso-belica/sumy.

[Habr/HeadHunter, 2022] Habr/HeadHunter (2022). Habr.com summarization is all you need. https://habr.com/ru/companies/hh/articles/698212/.

[Huyen, 2023] Huyen, C. (2023). Rlhf: Reinforcement learning from human feedback. https://huyenchip.com/2023/05/02/rlhf.html.

[Shariati, 2021] Shariati, S. (2021). Text summarization techniques. https://sinashariati.medium.com/text-summarization-techniques-a4c9aa4fcfdb.

[Skoghäll and Öhman, 2022] Skoghäll, T. and Öhman, D. (2022). Summarization and keyword extraction on customer feedback data. https://www.diva-portal.org/smash/get/diva2:1672022/FULLTEXT01.pdf.