Byte Pair Encoding — The Dark Horse of Modern NLP

A simple data compression algorithm first introduced in 1994 supercharging almost all advanced NLP models of today (including BERT).

Background

The last few years have been an exciting time to be in the field of NLP. The evolution from sparse frequency-based word vectors to dense semantic word representation pre-trained models like Word2vec and GloVe set the foundation for learning the meaning of words. For many years, they served as reliable embedding layer initializations to train models in the absence of

large amounts of task-specific data. Since the word embedding models pre-trained on Wikipedia were either limited by vocabulary size or the frequency of word occurrences, rare words like `athazagoraphobia` would never be captured resulting in unknown `<unk>` tokens when occurring in the text.

Dealing with rare words

Character level embeddings aside, the first real breakthrough at addressing the rare words problem was made by the researchers at the University of Edinburgh by applying subword units in Neural Machine Translation using Byte Pair Encoding (BPE). Today, subword tokenization schemes inspired by BPE have become the norm in most advanced models including the very popular family of contextual language models like BERT, GPT-2, RoBERTa, etc. Some referred to BERT as the beginning of a new era, yet, I refer to BPE as a dark horse in this race because it gets lesser attention (pun intended) than it deserves in the success of modern NLP models. In this article, I plan on shedding some more light on the details on how Byte Pair Encoding is implemented and why it works!

The Origins of Byte Pair Encoding

Like many other applications of deep learning being inspired by traditional science, Byte Pair Encoding (BPE) subword tokenization also finds its roots deep within a simple lossless data compression algorithm. BPE was first introduced by Philip Gage in the article "A New Algorithm for Data Compression" in the February 1994 edition of the C Users Journal as a technique for data compression that works by replacing common pairs of consecutive bytes with a byte that does not appear in that data.

Repurposing BPE for Subword Tokenization

To perform subword tokenization, BPE is slightly modified in its implementation such that the frequently occurring subword pairs are merged together instead of being replaced by another byte to enable compression. This would basically lead the rare word `athazagoraphobia` to be split up into more frequent subwords such as `['_ath', 'az', 'agor', 'aphobia']`.

Step 0. Initialize vocabulary.

Step 1. Represent each word in the corpus as a combination of the characters along with the special end of word token `</w>`.

Step 2. Iteratively count character pairs in all tokens of the vocabulary.

Step 3. Merge every occurrence of the most frequent pair, add the new character n-gram to the vocabulary.

Step 4. Repeat step 3 until the desired number of merge operations are completed or the desired vocabulary size is achieved (which is a

hyperparameter).

What makes BPE the secret sauce?

BPE brings the perfect balance between character- and word-level hybrid representations which makes it capable of managing large corpora. This behavior also enables the encoding of any rare words in the vocabulary with appropriate subword tokens without introducing any “unknown” tokens. This especially applies to foreign languages like German where the presence of many compound words can make it hard to learn a rich vocabulary otherwise. With this tokenization algorithm, every word can now overcome their fear of being forgotten (athazagoraphobia).

References

1. Philip Gage, *A New Algorithm for Data Compression*. “Dr Dobbs Journal”
2. Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
3. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

arXiv preprint arXiv:1810.04805.

4. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Machine Learning | NLP | Deep Learning | Artificial Intelligence

Written by Akashdeep Singh Jaswal

93 Followers · Writer for Towards Data Science

ML Engineer | I enjoy being uncomfortable and solving complex problems through smart (AI) solutions. https://www.linkedin.com/in/akashjaswal/

More from Akashdeep Singh Jaswal and Towards Data Science

Akashdeep Singh Jaswal in Towards Data Science

Imagining a world without Transformers—Single Headed...

Distilling key ideas from one of the most entertaining NLP papers picturing a world...

Jan 6, 2020 80 1

Shaw Talebi in Towards Data Science

5 AI Projects You Can Build This Weekend (with Python)

From beginner-friendly to advanced

Oct 9 3.3K 58

Mauro Di Pietro in Towards Data Science

GenAI with Python: Build Agents from Scratch (Complete Tutorial)

with Ollama, LangChain, LangGraph (No GPU, No APIKEY)

Sep 29 1.8K 24

Thuwarakesh Murallie in Towards Data Science

I Fine-Tuned the Tiny Llama 3.2 1B to Replace GPT-4o

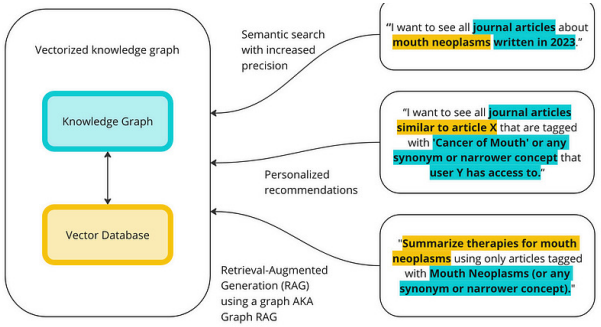
Is the fine-tuning effort worth more than few-shot prompting?

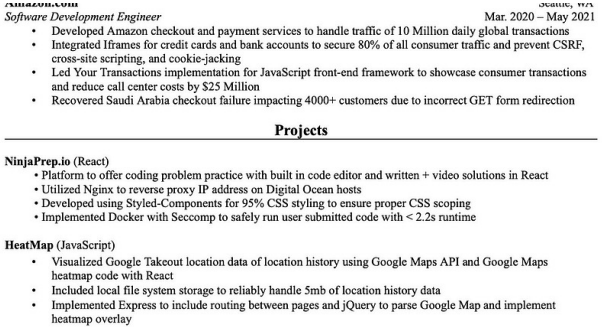
Oct 15 1.8K 22

See all from Akashdeep Singh Jaswal

See all from Towards Data Science

Recommended from Medium





Software Development Engineer Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by $25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects

NinjaPrep.io (React)
- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)
- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay

Steve Hedden in Towards Data Science

How to Implement Graph RAG Using Knowledge Graphs and...

A Step-by-Step Tutorial on Implementing Retrieval-Augmented Generation (RAG),...

Sep 6 1.4K 18

Alexander Nguyen in Level Up Coding

The resume that got a software engineer a $300,000 job at Google.

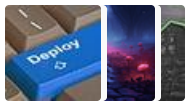
1-page. Well-formatted.

Jun 1 24K 486

Lists

Natural Language Processing

1768 stories · 1369 saves



Predictive Modeling w/ Python

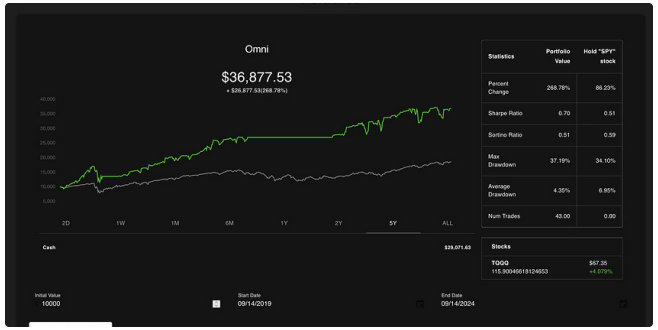
20 stories · 1612 saves

AI Regulation

6 stories · 595 saves

Practical Guides to Machine Learning

10 stories · 1964 saves



Austin Starks in DataDrivenInvestor

I used OpenAI's o1 model to develop a trading strategy. It is...

It literally took one try. I was shocked.

Sep 15 4.4K 120

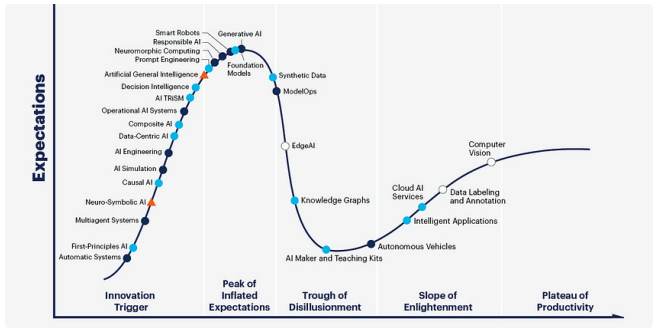
entity_group	score	word	start	end
AGE	0.541046	13 y.o.	11	19
SEX	0.549964	male	19	24
CLINICAL_EVENT	0.598481	seen	30	35
DISEASE_DISORDER	0.802787	PAH deficiency	53	68
SIGN_SYMPTOM	0.798482	vitamin D deficency	80	100
SIGN_SYMPTOM	0.571304	anemia	104	111
MEDICATION	0.789797	viatmin B1, B2	125	140
MEDICATION	0.871156	iron tablets	144	157

Alva Rani James, PhD

Reviewing NER models: DeBERTa versus BioBERT for electronic...

DeBERTa and BERT-based

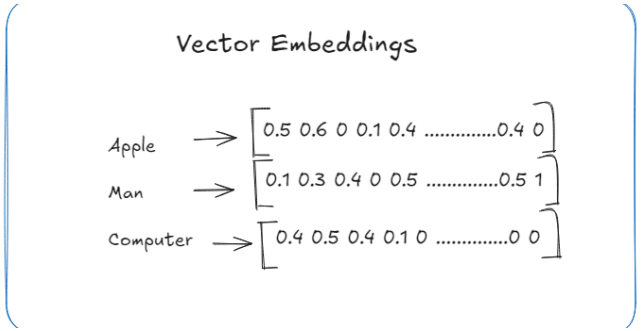
May 28 11



Vishal Rajput in AIGuys

Why GEN AI Boom Is Fading And What's Next?

Every technology has its hype and cool down period.



Mdabdullahalhasib in Towards AI

A Complete Guide to Embedding For NLP & Generative AI/LLM

Understand the concept of vector embedding, why it is needed, and...

Sep 4 2.4K 72

5d ago 78

See more recommendations