RAGged Edge Box: A private, single-user AI-based IR system

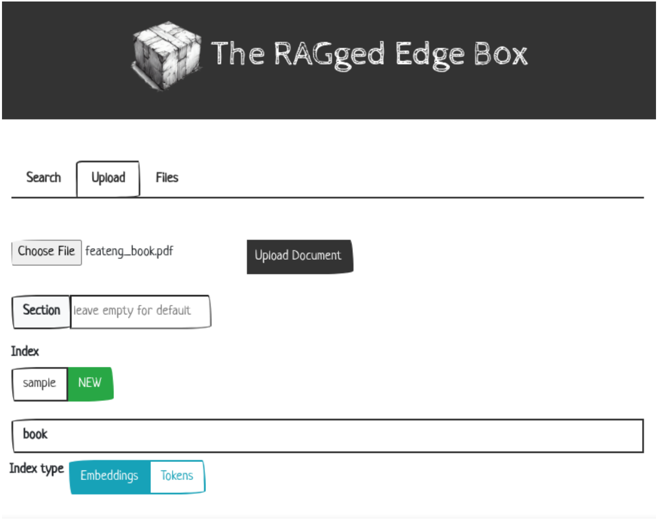
Pablo Duboue, PhD

Data Umbrella
Oct 2024

Project page

- https://textualization.com/ragged/
- Preview VM available for download

Screenshots: (1) Upload



feateng_book.pdf

Section: leave empty for default

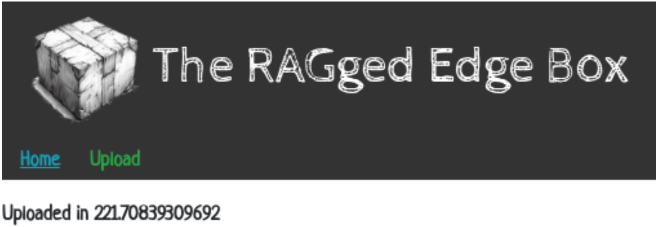
Index: sample | NEW

book

Index type: Embeddings | Tokens

Figure: Upload The Art of Feature Engineering

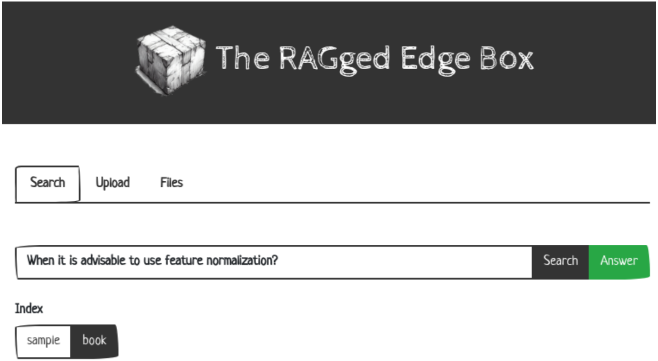
Screenshots: (2) Upload results



```
Uploaded in 221.70839309692
```

Figure: 4 minutes later. . .

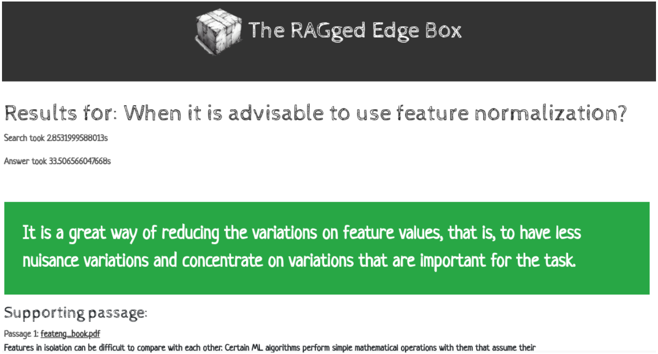
SCREENSHOTS: (3) QUERY



When it is advisable to use feature normalization?

FIGURE: When it is advisable to use feature normalization?

Screenshots: (4) Answered



Results for: When it is advisable to use feature normalization?

Search took 2.8531999588013s

Answer took 33.506566047668s

It is a great way of reducing the variations on feature values, that is, to have less nuisance variations and concentrate on variations that are important for the task.

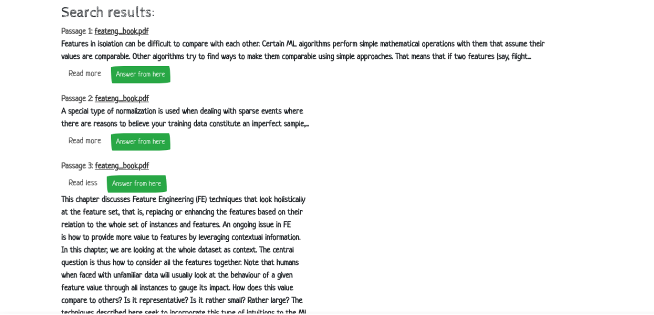
Supporting passage:

Passage 1: feateng_book.pdf

Features in isolation can be difficult to compare with each other. Certain ML algorithms perform simple mathematical operations with them that assume their

Figure: Half a minute

Screenshots: (5) Other passages



Search results:

Passage 1: feateng_book.pdf

Features in isolation can be difficult to compare with each other. Certain ML algorithms perform simple mathematical operations with them that assume their values are comparable. Other algorithms try to find ways to make them comparable using simple approaches. That means that if two features (say, flight...

Passage 2: feateng_book.pdf

A special type of normalization is used when dealing with sparse events where there are reasons to believe your training data constitute an imperfect sample,...

Passage 3: feateng_book.pdf

This chapter discusses Feature Engineering (FE) techniques that look holistically at the feature set, that is, replacing or enhancing the features based on their relation to the whole set of instances and features. An ongoing issue in FE is how to provide more value to features by leveraging contextual information. In this chapter, we are looking at the whole dataset as context. The central question is thus how to consider all the features together. Note that humans when faced with unfamiliar data will usually look at the behaviour of a given feature value through all instances to gauge its impact. How does this value compare to others? Is it representative? Is it rather small? Rather large? The techniques described here seek to incorporate this type of intuitions to the ML

Figure: Answer from here

Attendee personas

- [Sci] Scientist
  - Research outreach without hosting expensive servers
- [Usr] Potential user
  - Values privacy, don't want to pay server fees
- [Tech] Potential re-user
  - Likes the VM technology for a different stack
- [Tech$] Potential ISV
  - Likes to build (and sell) solutions on top
  - Knows PHP
- [Tech] Potential contributor
  - Values Free software
  - Interested in AI

This talk

1. RAGged Edge Box demo [User]
2. AI Concepts (RAG, LLMs, Embeddings) [User, Tech]
3. RAG Concepts (IR, chunk, prompt) [User, Tech]
4. RAGged Edge Box (concept, advantages) [User]
5. RAGged Edge Box Architecture [Tech]
6. Enabling Technology Bits (ONNX, PHP Semantic Search) [Sci, Tech]
7. Extension Points [Sci, Tech$]
8. VM Packaging [Tech]
9. RAGged Edge Box as a Platform [Tech$]

Outline

- AI Concepts [User, Tech]
- RAG Concepts [User, Tech]
- RAGged Edge Box [User]
- RAGged Edge Box Architecture [Tech]
- Enabling Technology Bits [Sci, Tech]
- Extending RAGged Edge Box [Sci, Tech]
- VM Packaging [Tech]
- RAGged Edge Box as a Platform [Tech]

What is RAG

- Retrieval Augmented Generation: combine LLMs with existing information

1. Retrieve information
   - Search engines, including embedding-based ones
2. Give it to the LLM as input
   - In the prompt
3. Ask the LLM to do things with that information
   - For example, to answer a question

What are LLMs

- Cellphone autocomplete is a language model
- It “knows” things
  - I really want to...
    - eat empanadas
    - learn about LLMs
- A large language model is similar but having seen tons of text
  - Some of those texts contain exams and their answers
  - Or directly “instructions” to make the LLM more useful

What are Embeddings

- Encode information by projecting it to a fixed space
- In the case of RAG and “semantic search” the input is a text span, the output is a fixed vector of floating point numbers (e.g., 384 numbers)
- The hope is that text conveying similar meaning will be represented by vectors close to each other in the Euclidean space

EMBEDDINGS ISSUES

- Many open questions
  - How to generate these embeddings?
    - Siamese neural networks
  - What type of semantic information is being captured?
  - How big is the span of text to make sense to compare the generated vectors?
  - What if one of the texts is short (like a question) and the other is long (like a paragraph)?
    - Asymmetric embeddings

Outline

AI Concepts [User, Tech]

RAG Concepts [User, Tech]

RAGged Edge Box [User]

RAGged Edge Box Architecture [Tech]

Enabling Technology Bits [Sci, Tech]

Extending RAGged Edge Box [Sci, Tech]

VM Packaging [Tech]

RAGged Edge Box as a Platform [Tech]

Answer extraction using LLMs

- LLMs can do many things (most of them badly)
- Answer extraction is one they are good at. Given the text:

  This text talks about many things. Among them is how 42 is a number featured in the Hitchhiker's Guide to the Galaxy, the comedy work by Douglas Adams.

- We can ask the LLM to extract the answer to the question "Why people are obsessed with the number 42?"

  People are obsessed with the number 42 because it is featured in the Hitchhiker's Guide to the Galaxy, a comedy work by Douglas Adams.

RAG vs 'Hallucinations'

- Asked GPT-4 "Why people are obsessed with the number 42?"

  The obsession with the number 42 largely comes from its significance in the popular science fiction novel "The Hitchhiker's Guide to the Galaxy" by Douglas Adams. In the story, a group of hyper-intelligent beings builds a supercomputer named Deep Thought to calculate the "Answer to the Ultimate Question of Life, the Universe, and Everything." After much anticipation, Deep Thought reveals that the answer is simply the number 42.

  Since the publication of the book, . . .

IR

- Information retrieval has been around since the 1950s
  - One of the reasons we wanted computers, to begin with
- GenAI is great but it hasn't improved IR as much
  - RAG hinges on good IR
  - If the information to answer the question is not retrieved, there is not much the LLM can do about it
  - That doesn't mean the LLM will admit defeat, getting it to say "I don't know" is a tall order
- It's the GenAI revolution, not the IR revolution

IR SOTA

- Currently we have IR systems using:
  - keyword search
  - complex search queries (keywords plus operators)
  - embeddings
- The best performance uses a combination of these approaches

Chunk

- The information provided to the LLM in a RAG system is usually a segment of the relevant document
  - This segment is called a "chunk" of the document
- The full document cannot be processed by the LLM that have a maximum processing size including the input information and the answer output
  - Local LLMs are between 400 words to 1500 words
  - Commercial LLMs available through APIs can process full books

Chunk Issues

1. Chunk size:
   - The size should be large enough to answer questions
   - But small enough to fit into the LLM input and be semantically coherent to produce viable embeddings
2. Multi-chunk processing
   - Provide multiple chunks to the LLM at once
   - That might exhaust the input and confuse the LLM
   - Some questions need information from multiple sources

Prompt

- How to structure the input is called the “prompt” of the LLM
- Different LLMs need different prompts
  - They can be sensitive to minuscule changes (like a carriage return character at the end of the prompt)

RAG Prompt Example

```
Use the following pieces of context to answer the
question  at the end. If you don’t know the answer,
just say that you don’t know, don’t try to make up
an answer.

{context}

Question: {question}
Answer:
```

Outline

- AI Concepts [User, Tech]
- RAG Concepts [User, Tech]
- RAGged Edge Box [User]
- RAGged Edge Box Architecture [Tech]
- Enabling Technology Bits [Sci, Tech]
- Extending RAGged Edge Box [Sci, Tech]
- VM Packaging [Tech]
- RAGged Edge Box as a Platform [Tech]

RAGged Edge Box

- RAGged Edge Box is a RAG system implemented as edge computing
  - There is no back-end, nothing runs “in the cloud”
- Self-contained virtual machine (3.0Gb) with a bare-bones linux set-up and the key components for RAG:
  - An embedding model and associated execution code
    - Sentence Distill Roberta (326Mb)
  - A local LLM and its associated server
    - Bling Stable-LM 3B Q4_K_M (1.71Gb)
  - A standalone search engine (keywords and embeddings)
    - SQLite v3 with VSS and FTS extensions
  - Web portal to upload documents, index them and query them
    - Custom made in PHP

Privacy

- Searching in a document collection should not involve giving access to the documents to a third party
- “Cloud” is a misnomer: it is just someone else’s computer

Technical Sovereignty

- Solutions relying on APIs hosted in other countries are not particular soverign
- Solutions relying on expensive hardware are not particular soverign

Against Planned Obsolescence

- Tired of tools that no longer work after a few months?
  - System Python upgrades break virtual environments
- VM life cycle independent of the operating system life cycle
  - Download a tool that will remain useful for years
- Dependencies packed in long terms storage solutions:
  - Docker
  - Debian stable
  - Composer

Outline

- AI Concepts [User, Tech]
- RAG Concepts [User, Tech]
- RAGged Edge Box [User]
- RAGged Edge Box Architecture [Tech]
- Enabling Technology Bits [Sci, Tech]
- Extending RAGged Edge Box [Sci, Tech]
- VM Packaging [Tech]
- RAGged Edge Box as a Platform [Tech]

Architecture

- `site/index.php`
- `site/upload.php`
- `site/search.php`

from PHP Semantic Search:

- `Embedder.php`
- `Index.php`
- `Tokenizer.php`

Outline

- AI Concepts [User, Tech]
- RAG Concepts [User, Tech]
- RAGged Edge Box [User]
- RAGged Edge Box Architecture [Tech]
- Enabling Technology Bits [Sci, Tech]
- Extending RAGged Edge Box [Sci, Tech]
- VM Packaging [Tech]
- RAGged Edge Box as a Platform [Tech]

ONNX

- https://onnx.ai/
- Open Neural Network Exchange
- Specify the neural network graph in a vendor-independent manner
- Train using any framework, execute using a different framework
    - Execute without large dependecies
- Open source ONNX runtime created and maintained by Microsoft
    - Runtime phones home without special parameters
- Packed for PHP by Andrew Kane: https://packagist.org/packages/ankane/onnxruntime

LLAMA.CPP

- `https://github.com/ggerganov/llama.cpp`
- Transformer implementation in C++
- High performance execution on CPU
- `llama-server` allows for local API calls
- Supports LLMs in `gguf` format
  - Allows for mixed execution in low-RAM GPUs

Why PHP

- Large installed code bases (Wikimedia, WordPress, Nextcloud)
- Used by nearly 80% of all websites
- Brainpower available worldwide
- A PHP install is simpler and smaller than a Python/Java/etc install
  - parsing/compilation/execution/cleanup is a functional paradigm. Side effects go to the DB
- 11th most popular programming language in 2023

PHP Semantic Search classes

- A subcontract with EvoluData, a Quebec company specialized in PHP solutions.
    - Funded through an AI innovation grant from the Quebec government
- `https://packagist.org/packages/textualization/semantic-search`
    - `KeywordIndex.php` Okapi BM25 implemented on top of SQLite3 text search
    - `VectorIndex.php` embedding search using SQLite3 Vector Search (FAISS) extension
- Support for HyDE and local LLM embeddings

Example

```
\Textualization\SemanticSearch\Ingester::ingest([
   "location"=>"index.db",
   "class"=>"\\Textualization\\SemanticSearch\\VectorIndex"
 ], [], "docs.jsonl");
```

Sentence Transformers (Embeddings)

- packagist.org/packages/textualization/sentence-transphormers
- packagist.org/packages/textualization/ropherta
- packagist.org/packages/textualization/ropherta-tokenizer

Sentence Transformers (Embeddings)

```
use \Textualization\SentenceTransphormers\SentenceRopherta;

$model = new SentenceRopherta();
$emb = $model->embeddings("Text");

// alt. using the semantic search classes

$e = \Textualization\SemanticSearch\SentenceTransphormerEmbedder();
$emb = $e->encode("Text");
```

Reverse Engineering Huggingface Components

- This was by far the most time consuming aspect of the work last year

Outline

- AI Concepts [User, Tech]
- RAG Concepts [User, Tech]
- RAGged Edge Box [User]
- RAGged Edge Box Architecture [Tech]
- Enabling Technology Bits [Sci, Tech]
- Extending RAGged Edge Box [Sci, Tech]
- VM Packaging [Tech]
- RAGged Edge Box as a Platform [Tech]

New IR

- Subclass `Index.php`
  - on GitHub

New LLM

- Put it in `download` and change the `box/launch-llama.sh` script to use it

Better document handling

- Change `site/upload.php`

Outline

- AI Concepts [User, Tech]
- RAG Concepts [User, Tech]
- RAGged Edge Box [User]
- RAGged Edge Box Architecture [Tech]
- Enabling Technology Bits [Sci, Tech]
- Extending RAGged Edge Box [Sci, Tech]
- VM Packaging [Tech]
- RAGged Edge Box as a Platform [Tech]

Generating Virtual Box images programmatically

- Project structure (on GitHub)
- `Dockerfile`
- `make-image.sh`

Outline

AI Concepts [User, Tech]

RAG Concepts [User, Tech]

RAGged Edge Box [User]

RAGged Edge Box Architecture [Tech]

Enabling Technology Bits [Sci, Tech]

Extending RAGged Edge Box [Sci, Tech]

VM Packaging [Tech]

RAGged Edge Box as a Platform [Tech]

Business

- My hope is that the project enables ISVs to adapt the PHP code for customer specific needs:
  - Specific document segmentation and detagging
  - Improved IR using faceted search
  - Handling additional file formats
  - Plugging in more performat IR engines (e.g., Manticore/MariaDB vector)
- Not all GenAI money should find its way back to Nvidia/Microsoft

Status

- Full automatic VM creation
- Missing functionality:
  - Deletion
  - Hybrid embeddings + keywords
  - Keywords most probably doesn't work due to chunk size
  - Upgrade
  - API

Multilinguality

- The original project for PHP Semantic Classes was in French
  - A complex tokenizer handling 100 languages (SentencePiece) was also migrated to PHP:
- https://packagist.org/packages/textualization/sentencepiece
- Multilingual embeddings (supporting English, Spanish, French +100 languages):
  - https://huggingface.co/intfloat/multilingual-e5-small
  - The current VM does not have these requirements installed

Multilinguality: Needs

- We need a small local LLM that can do multilingual answer extraction
  - Or at least in Spanish
  - Ideas?

Contributing to the Project

- github.com/Textualization/the-ragged-edge-box/
- textualization.com/ragged/
- Non-trivial PR will be listed as vetted ISVs

Other Announcements

- Apache UIMA-CPP
- GSoC
- llama.cpp Annotator: `https://github.com/Textualization/LlmAnnotator`

Conclusions

- It is time to go back to the P in NLP
  - Natural Language **Processing**
- Successful LLM deployments need a lot of programming and smarts outside the LLM bits
- The RAGged Edge Box project allows new players versed in traditional programming to join the field