Python Ireland

How I use Python to prepare for my astronomy podcast

Marcel-Jan Krijgsman

Introduction

Marcel-Jan Krijgsman

Senior Data Engineer for DIKW Intelligence

Working with Python since 2017

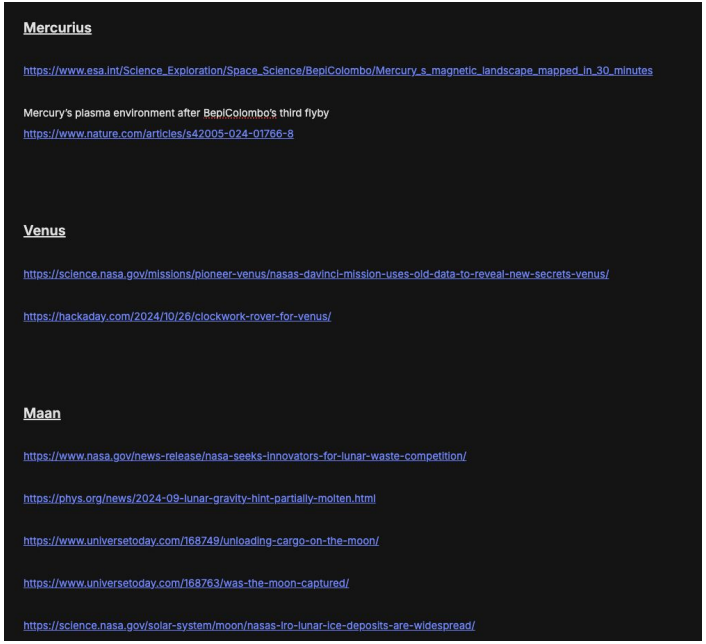
For work, but also for hobby.

Origin story

Since 2015 I do a half-yearly presentation on “recent developments in the solar system” for our astronomy group.

For this I follow a lot of astronomy news to follow all the news on planets, moons, asteroids and comets.

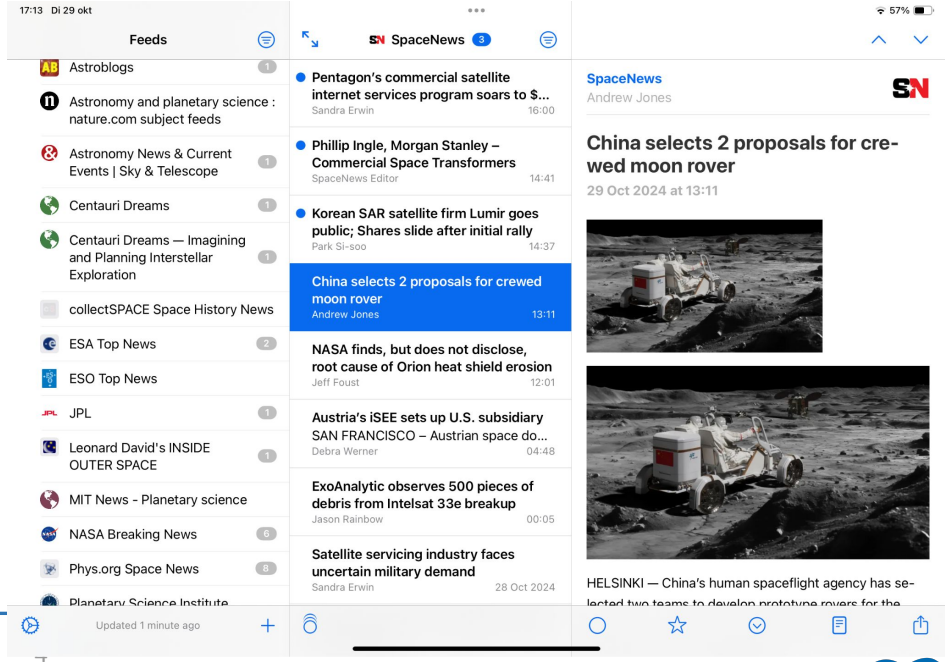
For this I keep a monthly Evernote document with links to all related articles.



Astronomy news overload

I’m following the RSS feeds of 20+ websites with an app.

And it is getting too much. I’m more copying links than actually reading the news.



What would help me

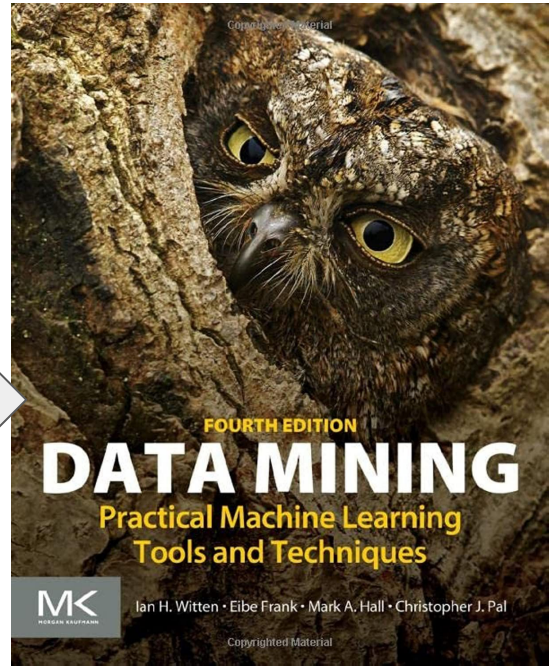
Is there a way to automate this? To pick up the news and put it into categories for me.

Isn’t there a thing called text mining?

I asked data scientists (at my company).

Stack Overflow said I needed to study this book

(654 pages!)



And then I thing called ChatGPT arrived

The least that you could say, is that it is good at text.

Can it do this work for me?

I’ve created prompts to pick a category and 5+ tags.

Let's ask ChatGPT to categorize news for me

```
astro categories = "Mercury, Venus, Moon, Earth, Mars, " \
                   "Jupiter, Saturn, Uranus, Neptune, " \
                   "Pluto and the Kuiper Belt, Comets, " \
                   "Exoplanets, Formation of the Solar System, " \
                   "Telescopes, Meteorites, " \
                   "Artificial Intelligence, Miscellaneous"

user followup = f"Categorize this text in one of the following categories:\n\n
{astro_categories}"
message_history.append({"role": "user", "content": f"{user_followup}"})

completion = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=message_history
)

reply_content = completion.choices[0].message.content
print(reply_content)
```

Results were mixed at first

Lowercase, uppercase, capitalized. If you don't tell it what to do, it will pick different things.

After a couple of months it decided capitalized meant all caps for some reason.

It came up with its own categories.

But wait: we're in Python. We can check its work.

Capitalization: Can be done afterwards.

Categories: Check the outcome and compare with the category list.

Prompt that works best for me

```
prompt = f"""
            You are given a title and a summary of a text. \
            The title is delimited by triple asterixes. \
            The summary is delimited by triple backticks. \
            ***{title}*** \
            ```{summary_text}``` \

            You are also given a list of topics. \
            List of topics: {astro_category_list} \

            Determine what is the main topic for this title and text. \
            If you have trouble finding a good main topic, \
            instead choose this topic: \
            Miscellaneous \
            The response should follow the format: \
            Main category: maintopic \
            and nothing else. \
            """
```
```

Embeddings

Other optimizations

Some websites repeat the articles from others.

“Hot news” is repeated a lot.

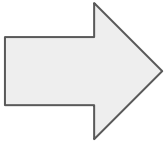
Do I want to use (pay for) ChatGPT multiple times to categorize the same content?

And that’s when I learned about embeddings.

Embeddings

Are the technology behind large language models.

“The Voyager 2 flyby of Uranus in 1986 revealed an unusually oblique and off-centred magnetic field. This single in situ measurement has been the basis of our interpretation of Uranus’s magnetosphere as the canonical extreme magnetosphere of the solar system; with inexplicably intense electron radiation belts and a severely plasma-depleted magnetosphere. However, the role of external forcing by the solar wind has rarely been considered in explaining these observations. Here we revisit the Voyager 2 dataset to show that Voyager 2 observed Uranus’s magnetosphere in an anomalous, compressed state that we estimate to be present less …”



```
[
    -0.057971809059381485,
    0.007631177082657814,
    -0.034021515399217606,
    0.10450427979230881,
    0.11810257285833359,
    -0.0483667217195034,
    0.006402068771421909,
    -0.08763448894023895,
    -0.01177206914871931,
    -0.01269279420375824,
    -0.05382048338651657,
    -0.08353287726640701,
    ..
]
```

Can embeddings detect if news is the same?

I've used Simon Willison's llm package.

With EMBEDDING_MODEL = "sentence-transformers/all-MiniLM-L6-v2" to create embeddings.

All my news articles are stored in Elasticsearch. So I've added embeddings to all news articles, as well to all new articles I'm finding.

Does it work?

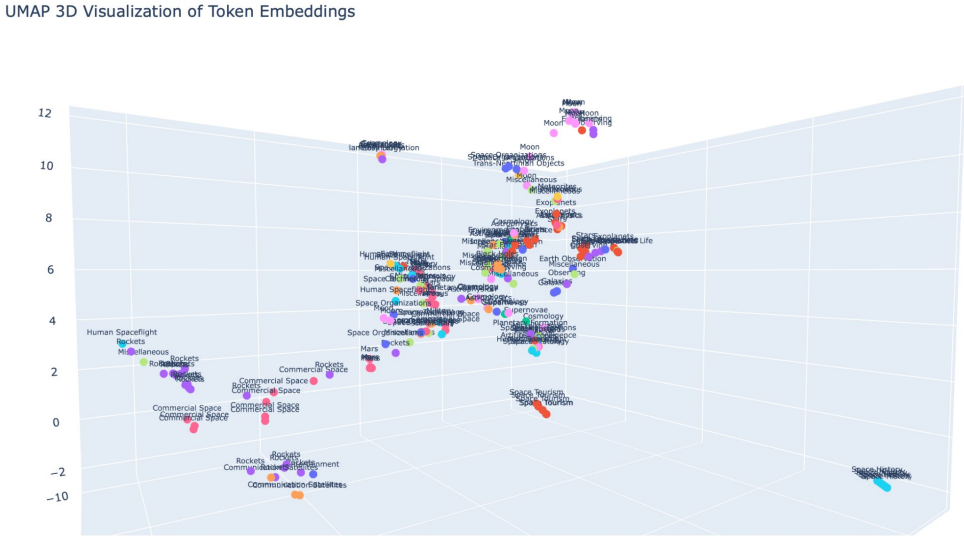
Not always, but it certainly is reusing ChatGPT results. Which saves money.

You can also visualize it

Flatten the 200-300 embedding dimensions to 3D.

I’ve tried PCA.

But I had the most success with UMAP



UMAP

```
        astronews_data = json.load(f)
        # Get cleaned_summary_vector from astronews_data

        cleaned_summary_vector = [newsitem['cleaned_summary_vector'] for newsitem in astronews_data]
        # Get main categories for labels.
        maincategories = [newsitem['universal_maincategory'] for newsitem in astronews_data]

        # Perform UMAP for dimensionality reduction (3D)
        umap_model = umap.UMAP(n_neighbors=3, min_dist=0.1, n_components=3)
        embeddings_umap = umap_model.fit_transform(cleaned_summary_vector)

        # Convert embeddings and labels to DataFrame for Plotly
        data_umap = {'x': embeddings_umap[:, 0], 'y': embeddings_umap[:, 1], 'z': embeddings_umap[:,
2], 'label': maincategories}
        df_umap = pd.DataFrame(data_umap)
```

A front-end app for retrieval

A front-end app to quickly find articles

Look, I'm not going to run Elasticsearch JSON queries every time I'm preparing a presentation.

I want something more user friendly.

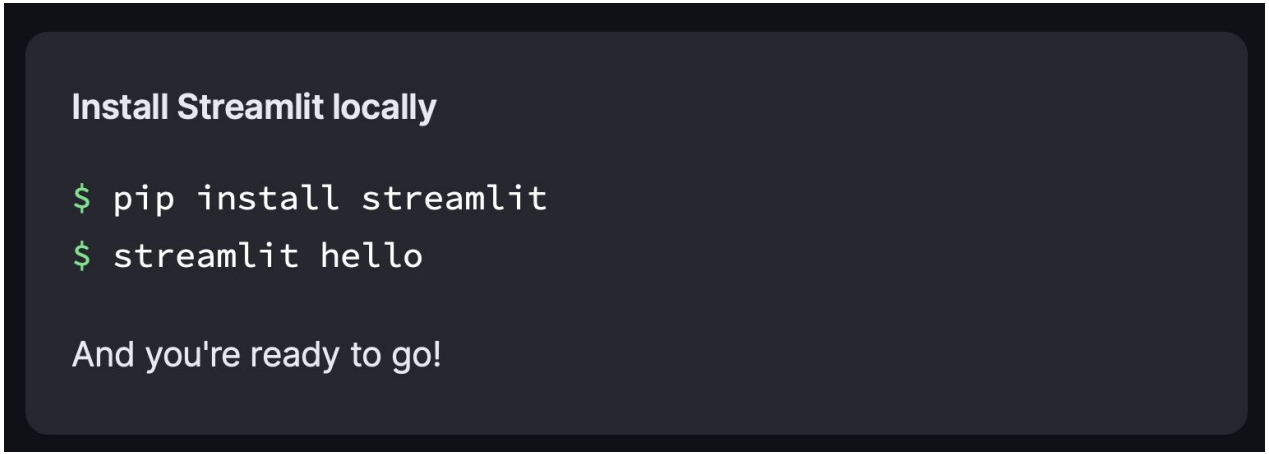
That would mean some kind of frontend application.

… and I have zero experience in that.

Enter Streamlit

A fast way to create simple frontends for data (https://streamlit.io/)

This is what they say:



What should be in my app

A way to select start date and end date. Preferably with some kind of calendar popping up.

A dropdown menu to select the category.

A tick box to select if I only want to get the “top categories” (which is something I’ve manually added to the data)

Is Streamlit really that easy?

Getting the selection stuff:

```
if __name__ == "__main__":
    st.title("Marcel-Jan's Astro Bulletin")
    # Add a default value for the start_date as one week ago
    start_date = st.date_input('Start date', value=(datetime.date.today()
                                                    - datetime.timedelta(days=7)))
    end_date = st.date_input('End date')

    maincategories = get_maincategories_from_elasticsearch(ELASTICSEARCH_CLIENT, \
                                                            ELASTICSEARCH_INDEX)
    maincategory = st.selectbox('Maincategory', maincategories)
    top_articles = st.checkbox('Top articles')
```

Showing the results

```
st.write('You selected:', maincategory)
st.write('Start date:', start_date)
st.write('End date:', end_date)
st.write('Top articles:', top_articles)

news_selection = get_news_selection_from_elasticsearch(ELASTICSEARCH_CLIENT, \
                                                       ELASTICSEARCH_INDEX)
news_selection_df = pd.DataFrame()

st.write(news_selection_df.to_html(escape=False, index=False), unsafe_allow_html=True)
```

Marcel-Jan's Astro Bulletin

Start date

2024/07/01

End date

2024/09/01

Maincategory

Moon

☐ Top articles

You selected: Moon

Start date: `2024-07-01`

End date: `2024-09-01`

Top articles: `False`

top_article	title	newsitem_datetime_str	astro_source_name	
False	NASA and JAXA exchange laser signals between SLIM lander and LRO in lunar orbit	2024-07-30 19:38:44	SpaceDaily.com	https://www.spacedaily.com/reports/NASA_and_JAXA_exchange_laser_signals_between_Lunar_surface_and
False	GMV advances Lunar rover navigation with	2024-07-30 19:38:44	SpaceDaily.com	https://www.marsdaily.com/reports/GMV_advances_Lunar_rover_navigation_with_FASTNAV_project_999.ht

But what about that podcast?

The podcast

I decided that if this would make my life easier, I might as well talk about “recent developments in the solar system” every month instead of every half year.

So why not start a podcast?

(Sorry, it’s in Dutch)

First episode: October 2023.

People are actually listening!

1370 unique listeners

5842 downloads

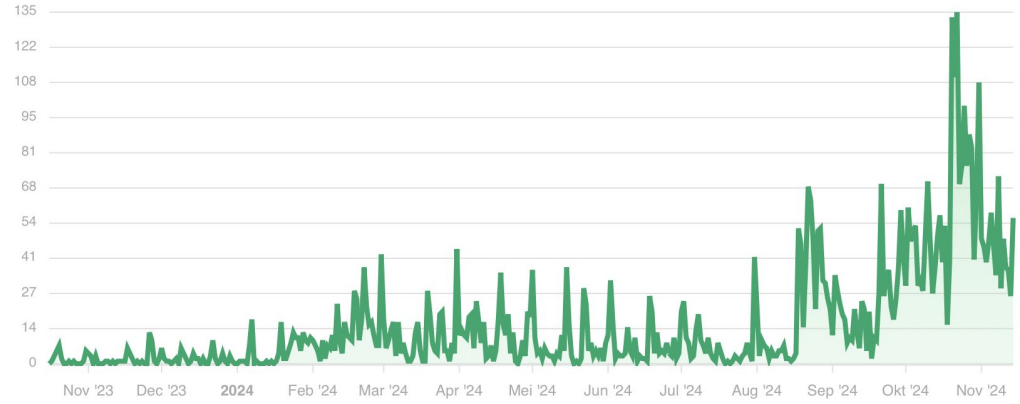
Amazing!

Downloads & unieke luisteraars

5842
Downloads

1370
Unieke luisteraars

Downloads



Het aantal downloads voor de geselecteerde periode is: **5842**.

Questions?