



RSS NEWS FEEDER PROJECT

NAFEEL JAVAID

465554

MUHAMMAD MUDASSAR ANWAR

4546925

TOUSEEF AKTHAR

460700

RAI AZMAT TARIQ

465039

ARBAB SAGIR

470382

Submitted by :Mudassar Anwar

RSS Feed Filter Code Explanation

Overview

This script fetches news items from RSS feeds (e.g., Google and Yahoo), parses them, and filters them based on user-defined triggers. The filtered news stories are then displayed in a GUI.

Libraries Used

feedparser: For parsing RSS feeds.

string: For string manipulation.

time, threading: For timing and concurrent execution.

project_util: For HTML translation (presumably defined elsewhere).

mtTkinter: A multi-threading friendly version of **Tkinter**.

datetime, pytz: For handling and converting date and time.

Code Breakdown

NewsStory Class

A data structure to hold information about a news story.

Attributes: guid, title, description, link, pubdate.

Methods to retrieve these attributes (get_guid, get_title, etc.).

Trigger Classes

Abstract class Trigger and various subclasses that determine if a news story meets certain criteria.

PhraseTrigger: Checks if a given phrase is in the text.

TitleTrigger: Checks if a phrase is in the title.

DescriptionTrigger: Checks if a phrase is in the description.

TimeTrigger: Abstract class for time-based triggers, storing time in EST.

BeforeTrigger: Fires if the story was published before a given time.

AfterTrigger: Fires if the story was published after a given time.

Composite Triggers:

NotTrigger: Inverts the trigger condition.

AndTrigger: Fires if both triggers fire.

OrTrigger: Fires if either trigger fires.

Functions

process(url)

Fetches and parses news items from the given RSS feed URL. Converts the publication date to a datetime object in GMT timezone. Returns a list of NewsStory instances.

filter_stories(stories, triggerlist)

Filters a list of NewsStory instances based on a list of triggers. Returns a list of stories for which any trigger in the list fires.

read_trigger_config(filename)

Reads a trigger configuration file and creates trigger objects as specified. Returns a list of triggers to be used for filtering news stories.

GUI

Uses **Tkinter** to create a GUI that displays filtered news stories.

The GUI includes a frame, a scrollbar, and a text widget for displaying news.

Continuously polls RSS feeds, filters stories, and updates the display.

Main Thread (main_thread(master))

Sets up the GUI and continuously polls RSS feeds, processes them, filters them based on triggers, and updates the GUI with the filtered stories.

Entry Point

Starts the GUI application and spawns the main thread for polling and filtering news stories.

Functions of the Code

Fetching and Parsing RSS Feeds:

The process function retrieves and parses RSS feeds from specified URLs, creating `NewsStory` objects.

Defining and Using Triggers:

Various trigger classes (**`PhraseTrigger`**, **`TitleTrigger`**, etc.) allow filtering of news stories based on title, description, and publication date.

Composite triggers (**`AndTrigger`**, **`OrTrigger`**, etc.) enable complex filtering logic.

Filtering News Stories:

The **`filter_stories`** function applies a list of triggers to a list of news stories, returning only the stories that meet the trigger conditions.

Reading Configuration:

The **`read_trigger_config`** function reads a configuration file to set up triggers, enabling easy modification of trigger conditions without changing the code.

Graphical User Interface:

The GUI, created using **`Tkinter`**, displays the filtered news stories, updating periodically based on new data from the RSS feeds.

Polling and Updating:

The **`main_thread`** function continuously polls the RSS feeds at specified intervals, processes the news stories, filters them using the triggers, and updates the GUI.

Example Usage

The script can be customized by creating a trigger configuration file specifying the triggers and their conditions.

Users can run the script to continuously monitor news feeds and display stories that match their interests.

By using this code, users can effectively filter and stay updated with news stories that match specific criteria, providing a personalized news feed experience.

Here's a detailed report of the provided code, explaining its structure, components, and functionality:

Code components:

def process(url):

```
    feed = feedparser.parse(url)
```

```
    entries = feed.entries
```

```
    ret = []
```

for entry in entries:

```
    guid = entry.guid
```

```
    title = translate_html(entry.title)
```

```
    link = entry.link
```

```
    description = translate_html(entry.description)
```

```
    pubdate = translate_html(entry.published)
```

try:

```
    pubdate = datetime.strptime(pubdate, "%a, %d %b %Y %H:%M:%S %Z")
```

```
    pubdate.replace(tzinfo=pytz.timezone("GMT"))
```

```
except ValueError:
```

```
    pubdate = datetime.strptime(pubdate, "%a, %d %b %Y %H:%M:%S %z")
```

```
    newsStory = NewsStory(guid, title, description, link, pubdate)
```

```
    ret.append(newsStory)
```

```
return ret
```

Data Structures

The NewsStory class represents a news story with attributes for GUID, title, description, link, and publication date.

```
class NewsStory:

    def __init__(self, guid, title, description, link, pubdate):

        self.guid = guid

        self.title = title

        self.description = description

        self.link = link

        self.pubdate = pubdate


    def get_guid(self):

        return self.guid


    def get_title(self):

        return self.title


    def get_description(self):

        return self.description


    def get_link(self):

        return self.link


    def get_pubdate(self):

        return self.pubdate
```

Trigger Mechanism

Abstract Trigger Class: Serves as the base class for all triggers.

```
class Trigger(object):

    def evaluate(self, story):

        raise NotImplementedError
```

Phrase Triggers:

Check if a specific phrase is present in the title or description of a news story.

PhraseTrigger: Checks if a phrase is in a given text.

```
class PhraseTrigger(Trigger):  
    def __init__(self, phrase):  
        self.phrase = phrase.lower()  
  
    def is_phrase_in(self, text):  
        text = text.lower()  
        for p in string.punctuation:  
            text = text.replace(p, ' ')  
        words = text.split()  
        phrase_words = self.phrase.split()  
        for i in range(len(words) - len(phrase_words) + 1):  
            if phrase_words == words[i:i + len(phrase_words)]:  
                return True  
        return False
```

TitleTrigger:

Inherits from PhraseTrigger and checks the title.

```
class TitleTrigger(PhraseTrigger):  
    def evaluate(self, story):  
        return self.is_phrase_in(story.get_title())
```

DescriptionTrigger:

Inherits from PhraseTrigger and checks the description.

```
class DescriptionTrigger(PhraseTrigger):
```

```
def evaluate(self, story):  
    return self.is_phrase_in(story.get_description())
```

Time Triggers: Compare the publication time of a news story with a specific time.

TimeTrigger:

Base class for time-based triggers.

```
class TimeTrigger(TTrigger):  
    def __init__(self, time_string):  
        est = pytz.timezone("EST")  
        self.time = est.localize(datetime.strptime(time_string, "%d %b %Y %H:%M:%S"))
```

BeforeTrigger: Fires if the news story was published before the specified time.

```
class BeforeTrigger(TimeTrigger):  
    def evaluate(self, story):  
        return story.get_pubdate().replace(tzinfo=pytz.timezone("EST")) < self.time
```

AfterTrigger: Fires if the news story was published after the specified time.

```
class AfterTrigger(TimeTrigger):  
    def evaluate(self, story):  
        return story.get_pubdate().replace(tzinfo=pytz.timezone("EST")) > self.time
```

Composite Triggers: Combine multiple triggers.

NotTrigger: Inverts the result of another trigger.

```
class NotTrigger(TTrigger):  
    def __init__(self, trigger):  
        self.trigger = trigger
```



```
def evaluate(self, story):  
    return not self.trigger.evaluate(story)
```

AndTrigger: Combines two triggers with a logical AND.

```
class AndTrigger(Trigger):  
    def __init__(self, trigger1, trigger2):  
        self.trigger1 = trigger1  
        self.trigger2 = trigger2  
  
    def evaluate(self, story):  
        return self.trigger1.evaluate(story) and self.trigger2.evaluate(story)
```

OrTrigger: Combines two triggers with a logical OR.

```
class OrTrigger(Trigger):  
    def __init__(self, trigger1, trigger2):  
        self.trigger1 = trigger1  
        self.trigger2 = trigger2  
  
    def evaluate(self, story):  
        return self.trigger1.evaluate(story) or self.trigger2.evaluate(story)
```

Filtering Mechanism

The `filter_stories` function filters a list of `NewsStory` instances based on a list of triggers.

```
def filter_stories(stories, triggerlist):  
    filtered_stories = []  
    for story in stories:  
        for trigger in triggerlist:  
            if trigger.evaluate(story):  
                filtered_stories.append(story)
```

```
        break
```

```
    return filtered_stories
```

User-Specified Triggers

The `read_trigger_config` function reads a configuration file to create and configure triggers as specified.

```
def read_trigger_config(filename):
    trigger_file = open(filename, 'r')
    lines = []
    for line in trigger_file:
        line = line.rstrip()
        if not (len(line) == 0 or line.startswith('//')):
            lines.append(line)
    trigger_file.close()

    triggers = {}
    triggerlist = []

    for line in lines:
        parts = line.split(',')
        if parts[0] == "ADD":
            for name in parts[1:]:
                triggerlist.append(triggers[name])
        else:
            name = parts[0]
            trigger_type = parts[1]
            if trigger_type == "TITLE":
                triggers[name] = TitleTrigger(parts[2])
            elif trigger_type == "DESCRIPTION":
```

```

        triggers[name] = DescriptionTrigger(parts[2])
    elif trigger_type == "BEFORE":
        triggers[name] = BeforeTrigger(parts[2])
    elif trigger_type == "AFTER":
        triggers[name] = AfterTrigger(parts[2])
    elif trigger_type == "NOT":
        triggers[name] = NotTrigger(triggers[parts[2]])
    elif trigger_type == "AND":
        triggers[name] = AndTrigger(triggers[parts[2]], triggers[parts[3]])
    elif trigger_type == "OR":
        triggers[name] = OrTrigger(triggers[parts[2]], triggers[parts[3]])

    return triggerlist

```

Graphical User Interface

The `main_thread` function sets up and runs the Tkinter GUI. It periodically fetches stories, filters them, and updates the GUI with the filtered stories.

```

SLEEPTIME = 120 # seconds -- how often we poll

```

```

def main_thread(master):
    try:
        t1 = TitleTrigger("election")
        t2 = DescriptionTrigger("Trump")
        t3 = DescriptionTrigger("Clinton")
        t4 = AndTrigger(t2, t3)
        triggerlist = [t1, t4]

        frame = Frame(master)
        frame.pack(side=BOTTOM)

```

```

scrollbar = Scrollbar(master)
scrollbar.pack(side=RIGHT, fill=Y)

t = "Google & Yahoo Top News"
title = StringVar()
title.set(t)

ttl = Label(master, textvariable=title, font=("Helvetica", 18))
ttl.pack(side=TOP)

cont = Text(master, font=("Helvetica", 14), yscrollcommand=scrollbar.set)
cont.pack(side=BOTTOM)

cont.tag_config("title", justify='center')

button = Button(frame, text="Exit", command=root.destroy)
button.pack(side=BOTTOM)

guidShown = []

def get_cont(newstory):
    if newstory.get_guid() not in guidShown:
        cont.insert(END, newstory.get_title() + "\n", "title")
        cont.insert(END, "\n-----\n", "title")
        cont.insert(END, newstory.get_description())
        cont.insert(END

```

Overall Assessment

The code provides a robust and flexible solution for monitoring and filtering news stories based on specific criteria. It combines various Python libraries and concepts effectively to achieve its goals. The modular design ensures that each component is responsible for a specific task, making the code maintainable and extendable. By leveraging user-defined triggers, the application can adapt to different requirements and preferences, demonstrating a practical application of event-driven programming and real-time data processing.

```
C:\WINDOWS\system32\cmd.  X  +  v

Enter keywords (comma-separated): iran
Polling...
No keywords provided. Continuing to poll...
```

RSS Feed Filter

Google & Yahoo Top News

Thousands attend funeral prayers of President Ebrahim Raisi, others - Geo News

Thousands attend funeral prayers of President Ebrahim Raisi, others Geo NewsIranians pay last respect
s to late president Raisi killed in helicopter crash DAWN.comIran's Khamenei leads prayers at Raisi me
morial before tens of thousands Al Jazeera EnglishIran helicopter crash: Iranian president's funeral cere
mony to begin BBC.comThe death of the president changes the power dynamics in Iran The Economist

Breaking News | Iran Supreme Leader's Meeting with Important Personalities | Geo News - Geo News

Breaking News | Iran Supreme Leader's Meeting with Important Personalities | Geo News Geo NewsPM
Shehbaz attends memorial for President Raisi in Tehran The Express TribunePM Leaves For Pakistan A
fter Attending Commemoration Ceremony Of Late Iranian President UrduPoint NewsPM leaves for hom
eland after completing Iran's visit Radio PakistanAyatollah Khamenei Calls Ties With Pakistan Important,
Urges Removal Of Obstacles - Iran Front Page Iran Front Page - IFP News

Iran's Supreme Leader and Hamas chief at funeral ceremony for Iranian president

Expert provides insight into suspected cause of helicopter crash that killed Iran's president

Exit

```
C:\WINDOWS\system32\cmd.  X  +  v

Enter keywords (comma-separated): pakistan
Polling...
No keywords provided. Continuing to poll...
|
```

Google & Yahoo Top News

Rain washes out England-Pakistan T20 opener | Arab News PK - Arab News Pakistan

Rain washes out England-Pakistan T20 opener | Arab News PK Arab News PakistanWashout at Heading
ley denies England, Pakistan valuable World Cup prep ESPNcricinfoLIVE: England vs Pakistan – first T2
0 international cricket match Al Jazeera English[Watch] ENG Vs PAK 1st T20I Under Threat As Rain Dre
nches Headingley Park OneCricketPak Vs England T20 washed out as weather warnings issued in Leed
s Pakistan Today

Energy-starved Pakistan is seeking sanction-free Iranian gas, says Musadik Malik - Business Recorder

Energy-starved Pakistan is seeking sanction-free Iranian gas, says Musadik Malik Business RecorderPT
V's Official Web Portal PTV's Official Web PortalMusadik Malik emphasizes adoption of environment-fri
endly technology for sustainable development Dunya NewsDollar's rate in interbank market - Markets Bu
siness RecorderGovt to enhance production of petroleum products: Musadik Radio Pakistan

Breaking News | Iran Supreme Leader's Meeting with Important Personalities | Geo News - Geo News

Breaking News | Iran Supreme Leader's Meeting with Important Personalities | Geo News Geo NewsPM
Shehbaz attends memorial for President Raisi in Tehran The Express TribunePM Leaves For Pakistan A
fter Attending Commemoration Ceremony Of Late Iranian President UrduPoint NewsPM leaves for hom
eland after completing Iran's visit Radio PakistanAyatollah Khamenei Calls Ties With Pakistan Important,

Exit