PyTeaser

PyTeaser is a python app that can be fed any news article or app and extracts a text summary from it. The summary is generated based upon ranking each sentence throughout the article based on their relevancy to the entire piece, then using the top 5 sentences to produce the extract. The sentences are ranked based upon their relevance to the title, relevance to article keywords, position of the sentence and length of the sentence.

Platform

As this is a python app, PyTeaser can be run on a very wide range of platforms. Including Linux, Windows and MacOS. Python is also supported by many different IDE's. Allowing the project to be modified and used on almost any system. To build the platform I used both Windows and Linux with the VSCode IDE. On both Linux and Windows, the package can be built with Python2.7 installed and using the command `pip install pyteaser`. This command installs the package dependencies necessary to run the package. The package can also be installed by downloading the source, and once in the root directory running `python setup.py build` and `python setup.py install`.

```
PS C:\Workspace\University\Year3\SWEN301\assignment2\SWEN301-Project-2> python .\tests.py
..
----------------------------------------------------------------------
Ran 2 tests in 9.769s

OK
PS C:\Workspace\University\Year3\SWEN301\assignment2\SWEN301-Project-2> python
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from pyteaser import SummarizeUrl
>>> url = 'https://www.stuff.co.nz/business/better-business/103828573/scaffolding-company-owed-ird-2-million-of-taxes'
>>> summaries = SummarizeUrl(url)
>>> print summaries
[u'\n\nCompany director Chris Warren said they\xa0began operating in February and\xa0were\xa0committed to building a qua
lity business and being a good corporate citizen.', u'READ MORE:\n\n *\xa0Scaffolding employees rehired after business w
ent into liquidation\n\n *\xa0Director of scaffolding company facing tax-related charges\n\n *\xa0Scaffolding company pu
t in liquidation six days before Christmas\n\nPeri\xa0was the\xa0director of the New Plymouth businesses, both\xa0placed
 in\xa0liquidation on December 19 by the Commissioner of the Inland Revenue Department (IRD), and in voluntary administr
ation on December 2.', u'At an earlier appearance in March, he pleaded guilty to the offending which\xa0related to\xa0ta
x avoidance for businesses\xa0Chain Rigging and Scaffolding Limited (CRSL), a division of Chain Resources Limited (CRL).
', u'CRSL became a listed company on 27 September 2013.', u'IRD also filed a claim in relation to the same outstanding t
axes for CRL to amount of $475,322,\xa0which included a\xa0preferential claim of $303,952.']
>>>
```

Figure 1: Testing and running PyTeaser

In Figure 1, I gave PyTeaser a recent stuff article and it was able to produce a short summary of the article. In the state it is, PyTeaser does not output nicely formatted text.

History

PyTeaser is based upon a Scale project called textteaser. The original project was added to git in October 2013 but not much is known about the project before that. PyTeaser was created in November of 2013 by Xiao Xu. There are many python projects that are direct ports of the original textteaser, but PyTeaser was only based upon it. Xiao Xu continues to manage the project, routinely accepting merge requests made by members of the community. The original project, textteaser, was developed to combine the power of natural language processing and machine learning into a summarisation algorithm that can produce concise, helpful summaries of any article given.

Domain

PyTeaser operates in the domain of journalism and news reporting. Because of this, the articles given must be in a webpage format and have sufficient content to produce the data required for an accurate summarisation. As the package is python bases, there are view limitation to the type of system that it can be run on. Allowing it to make full use of system resources to generate summarisation quickly and accurately. In website articles, it is common for the article to be filled out with unnecessary words that only add to the bulk of the article, rather than having any value to the content. These words can throw off the algorithm as it is based upon word relevancy. To manage this, the package has a method of blocking these. Although it isn't perfect, it's a step in the right direction in terms of producing high quality results. The uses for this package come from users that require concise snippets of articles that are accurate enough to determine the premise of the article, a use case for this package could be a report/study database, where they can automatically generate a summary to describe the article for the uses of search, or for a user to decide whether the article is relevant to read. This could be applied to a law firm, where large numbers of documents are needed to be scanned for relevancy or a research company that needs relevant articles to support their research.

Component Architecture

The existing architecture is very basic and does not take advantage of sperate modules and classes that are supported by python. Each component is represented by a series of different methods in a single python file. The separate components that exist are:

- Summariser
    - Responsible for combining the rest of the package
- Link Parser
    - Converts a website link to a usable format
- Text Parser
    - Splits the article text into usable parts
- Scorer
    - Probably the largest component, scores different parts of the text based on metadata such as frequency
- Ranker
    - Ranks each parts of the text to determine the output summarisation based on certain criteria

In such a small and simple system, for the package to achieve its goal all components are required except for the URL parser, and there for the goose package. Goose is an external python library that convert a webpage into useable data such and sanitised text and html meta tags. This is information that can be used to help score and rank the article. PyTeaser follows a very linear path. It is very much written in a scripting manner rather than implementing an effective architecture for the design. Figure 2 demonstrates this linear progression

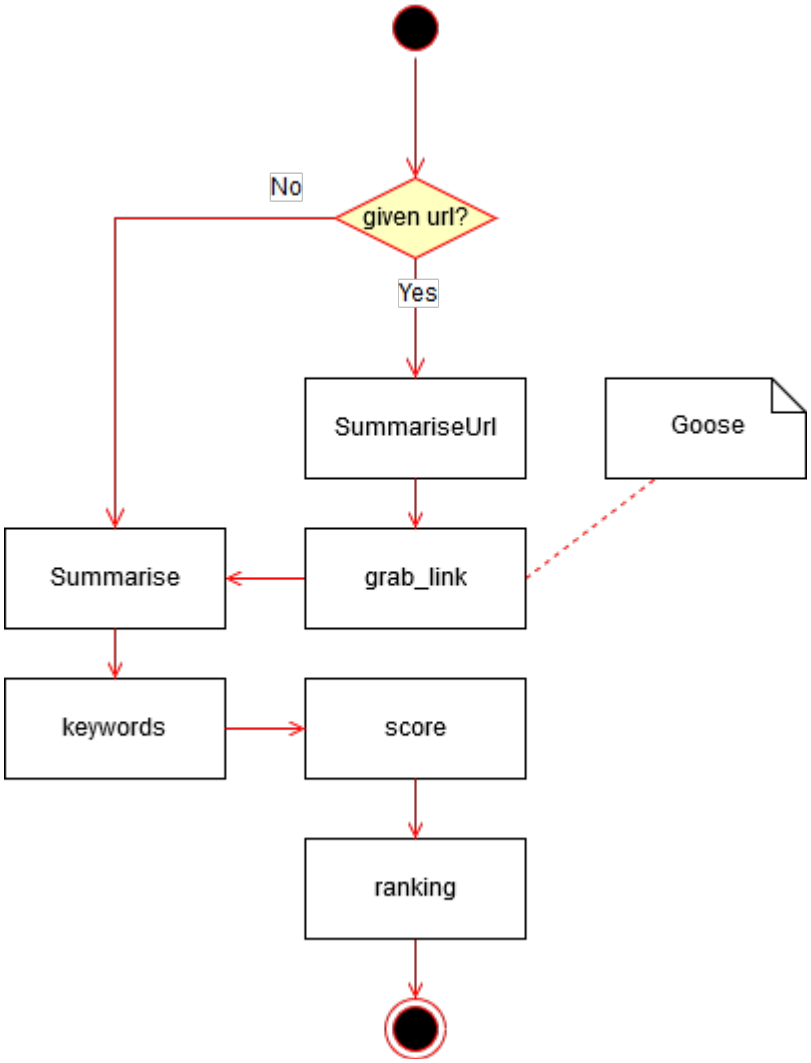

Figure 2: Activity Diagram

Data Structures

Python Version

PyTeaser is currently written using Python2. This is considered a legacy version of Python which is not compatible with Python3+[CITATION Wic17 \l 5129]. Python2 is no longer in active development and is considered to by at the end of its life. Given that Python3 was released in 2008, it is likely that the decision to stay on Python2 was due to library requirements. But this is no longer the case as each of the required libraries is now supported on Python3.

Testability

References

[1] M. Wichmann, “Python2 or Python3,” 10 09 2017. [Online]. Available: https://wiki.python.org/moin/Python2orPython3.