"Кто владеет информацией, тот владеет миром" Натан Ротшильд

Разработка системы извлечения знаний из гетерогенных источников данных для повышения качества принятия решений.

Система сбора и обработки информации с новостных и информационных порталов.

Толенбеков Ернур Канатбекович
Международный Университет Информационных Технологий,
Алматы Казахстан

План

- Введение
- Постановка задачи
- Алгоритм решения
  - Сбор данных
  - Предобработка данных
  - Оценка тональности
- Результаты
- Источники

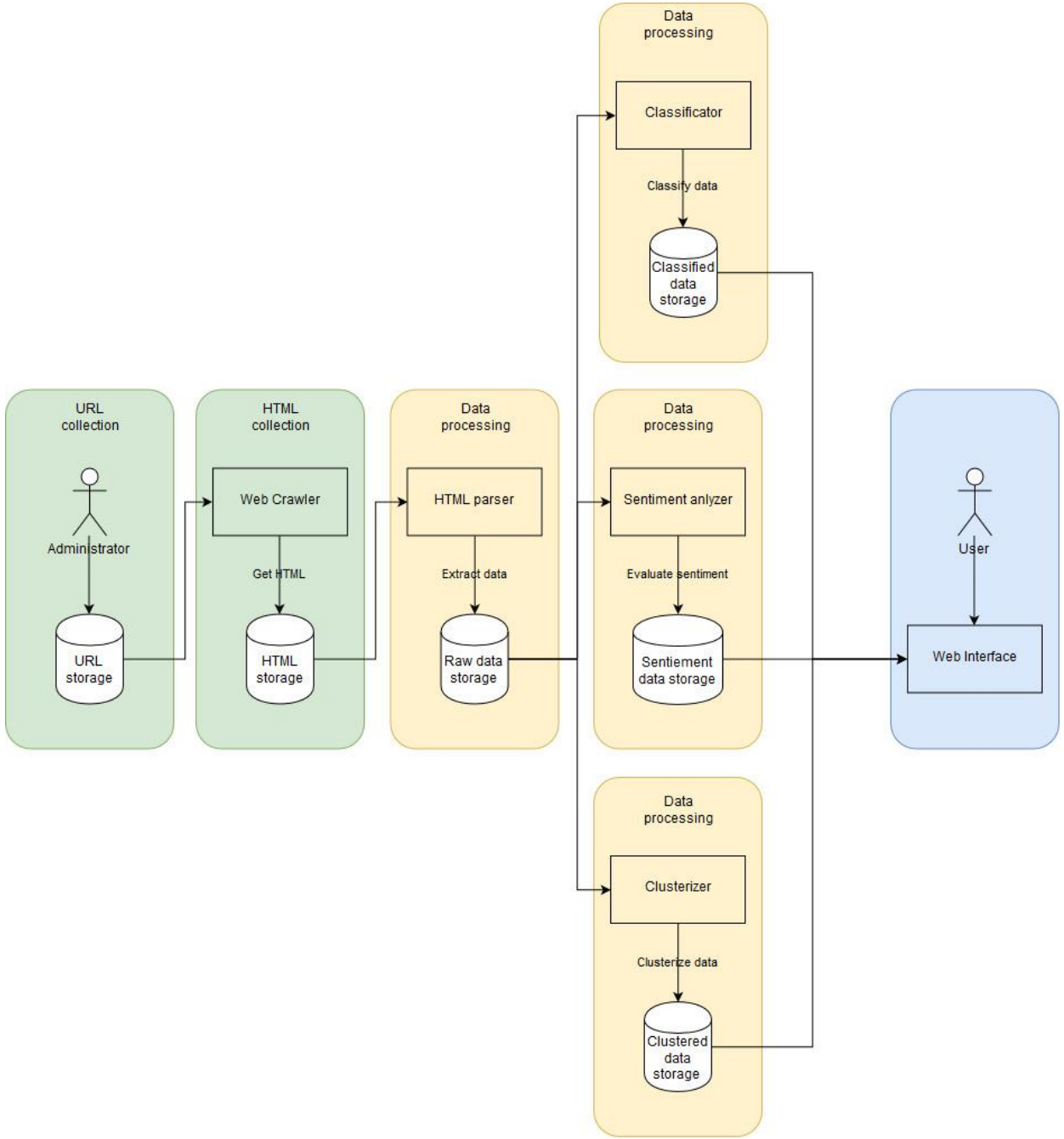
Введение

С бурным развитием информационных технологии в последние десятилетия количество генерируемой информации с каждым днем растет в экспоненциальном порядке. В былые времена человек не имел столь много ресурсов, откуда бы он мог брать нужную ему информация. Но сейчас, используя всемирную сеть, можно найти любую информация без каких либо усилии и затрат. В связи с этим, искать, обрабатывать, анализировать нужную информацию и быть осведомленным о происходящих событиях является важным инструментом для фильтрации потока информации по соответствующим запросам и интересам.

Постановка задачи

Для решения задачи информационная система должна была уметь решать следующие задачи:

- Искать и хранить информацию с разных источников данных
- Объединять информацию из разных источников в зависимости от события
- Определять реакцию (положительное, отрицательное, нейтральное) людей на определенное событие
- Предоставлять анализ по определенной тематике или новости



Архитектура системы

Алгоритм решения. Сбор данных

Python 2.7 библиотека Scrapy 1.4.0

- широкий спектр для настройки и управление краулером.
- не делает большую нагрузку на сторонние сервера, что в свою очередь уменьшает быть заблокированным.
- соблюдены все правила для ботов robots.txt

Сбор данных

возможности **Scrapy**:

- поиск и извлечение данных их HTML и XML
- преобразование данных перед экспортом
- экспорт в форматы JSON, CSV, XML
- скачивание файлов
- расширение фреймворка собственными middlewares, pipelines
- выполнение POST запросов, поддержка куков и сессий, аутентификации
- подмена user-agent
- shell консоль для отладки
- система логирования
- мониторинг через Web-интерфейс
- управление через Telnet-консоль

Сбор данных

- При краулинге для каждого сайта создавался отдельный "паук", в котором были прописаны правила парсинга отдельных страниц. Первым делом, краулер находит карту сайта (sitemap.xml) и выбрает нужные ссылки для загрузки. Далее грузится HTML документ страницы. После того как краулер загрузил HTML, весь полученный контент передается парсеру. Для точного определения нужного элемента в DOM, парсер используется xpath.
- При запуске модуля краулинга выбирается соответствующий "паук" для парсинга страниц. После того как парсер заканчивает обработку полученного текста, все данные записываются в базу данных.

Сбор данных. Структура sitemap

XML Sitemap

You can find more information about XML sitemaps on sitemaps.org.

This XML Sitemap Index file contains 198 sitemaps.

Sitemap	Last Modified
https://tengrinews.kz/sitemap-other.xml	2018-07-05 02:00
https://tengrinews.kz/sitemap-news-197.xml	2018-07-05 02:00
https://tengrinews.kz/sitemap-news-196.xml	2018-07-05 02:00
https://tengrinews.kz/sitemap-news-195.xml	2018-07-05 02:00
https://tengrinews.kz/sitemap-news-194.xml	2018-07-05 02:00

```
name = 'tengri_sitemap'
sitemap_urls = ['https://tengrinews.kz/sitemap-index.xml']
sitemap_rules = [
    ('/travel/', 'parse_alternate'),
    ('/mixnews/', 'parse_alternate'),
    ('/sitemap-other','no_parse'),
    ('', 'parse')
]
```

XML Sitemap

You can find more information about XML sitemaps on sitemaps.org.

This XML Sitemap contains 664 URLs.

↑ Sitemap Index

URL	Prio	Images	Ch. Freq.	Last Mod.
https://tengrinews.kz/strange_news/prinimayte-dush-vdvoem-kommunalschiki-samaryi-348313/	70%	1	Monthly	2018-07-05
https://tengrinews.kz/events/40-chelovek-tushat-pojar-v-tsentre-atyirau-348312/	70%	1	Monthly	2018-07-05
https://tengrinews.kz/kazakhstan_news/nazarbaev-nazval-lyubimoe-mesto-v-astane-348311/	70%	1	Monthly	2018-07-05
https://tengrinews.kz/events/neskolko-mikrorayonov-almatyi-ostalis-348309/	70%	1	Monthly	2018-07-05
https://tengrinews.kz/events/sanatoriy-gorel-v-turkestanskoy-oblasti-348308/	70%	1	Monthly	2018-07-04
https://tengrinews.kz/strange_news/Siri-sluchayno-sdelala-doklad-parlamente-348307/	70%	1	Monthly	2018-07-05

Алгоритм решения. Предобработка данных

- Перед тем как приступить непосредственно к самой классификации, текст проходит предобработку. Если не производит предобработку текста, то два почти одинаковых текста могут быть классифицированы по-разному. Вследствие чего результаты классификации могут оказаться неудовлетворительными.
- Процесс предобработки проходит в три этапа:
  - удаление стоп слов,
  - токенизация
  - стемминг.

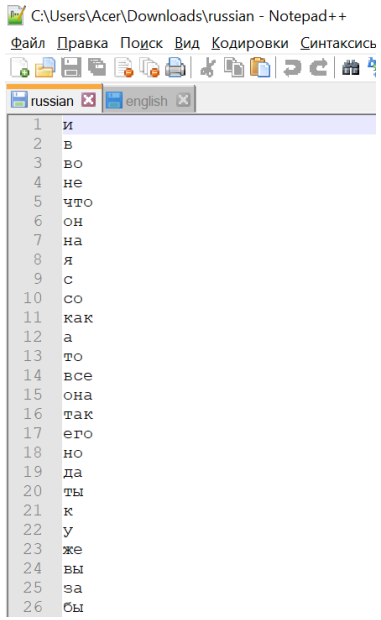
Предобработка данных

- pattern = r'(RT)*[@][\w]+:*‘ pattern_numbers = r'[0-9]*‘ pattern_symbols = r'(\W)(?=\1)'
- В первом этапе производилась токенизация текста, при котором текст делится на отдельные слова. Так как пробел или знаки препинания не всегда означают конец слова (например "Mr. Bond"), для реализации использовался модуль токенизации библиотеки nltk. В процессе токенизации символы, которые могут обозначать эмоции, считались как отдельные слова.

```
>>> from nltk.tokenize import TweetTokenizer
>>> tknzr = TweetTokenizer()
>>> s0 = "This is a cooool #dummysmiley: :-) :-P <3 and some arrows < > -> <--"
>>> tknzr.tokenize(s0)
['This', 'is', 'a', 'cooool', '#dummysmiley', ':', ':-)', ':-P', '<3', 'and', 'some', 'arrows', '<', '>', '->', '<--']
```

Предобработка данных

- В следующем этапе, удаляются слова не несущие смысловую нагрузку, так называемые стоп-слова. Например, из текста можно удалить слова "are, "aren't", "as". В качестве словаря стоп слов использовались словари из корпусов nltk.



Предобработка данных

- Полученный набор токенов проходил через процесс стемминга. При стемминге слова приводятся в начальную форму. Это нужно для того, чтобы различные словоформы одного слова были приведены к единому виду. Например, слова "jumped" и "jumping" после стемминга становятся словом "jump".

Here is a sample of Russian vocabulary, with the stemmed forms that will be generated with this algorithm.

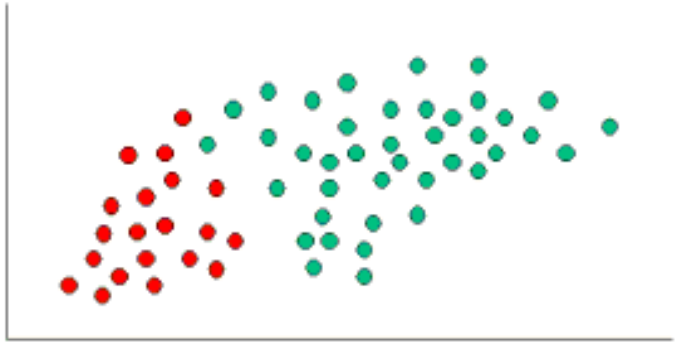
word	stem	word	stem
в	в	п	п
вавиловка	вавиловк	па	па
вагнера	вагнер	пава	пав
вагон	вагон	павел	павел
вагона	вагон	павильон	павильон
вагоне	вагон	павильонам	павильон
вагонов	вагон	павла	павл
вагоном	вагон	павлиний	павлин
вагоны	вагон	павлиньи	павлин
важная	важн	павлиньим	павлин
важнее	важн	павлович	павлович
важнейшие	важн	павловна	павловн
важнейшими	важн	павловне	павловн
важничал	важнича	павловной	павловн
важно	важн	павловну	павловн
важного	важн	павловны	павловн
важное	важн	павловцы	павловц
важной	важн	павлыч	павлыч

Алгоритм решения. Оценка тональности

- Есть множество методов и алгоритмов для анализа тональности, и каждый из них показывает хорошие результаты в зависимости от поставленной задачи и предметной области данных. Самые распространенные из них относится или к классификации с использованием словарей с заранее отмеченными коэффициентами тональности, или к классификации с использованием заранее размеченных кусков текста, которые используются в качестве тренировочного набора.
- Для нашей системы мы использовали наивно-байессовский метод, использующий тренировочный и тестовый набор. Выбор этого методы обосновывается тем, что в большинстве случаев мы анализируем тональность комментариев оставленных к новостям. Наивно-байессовкий классификатор предполагает, что слова в тексте очень слабо связаны между собой или же между ними нет никакой связи. Именно такими являются комментарии. В них мысли пользователя могут быть не структурированными, могут перескакивать от одной темы к другому и т.д.

Оценка тональности. Naïve Bayes Classification

Чтобы продемонстрировать концепцию классификации наивных байесов, рассмотрим приведенный ниже пример



Как указано, объекты могут быть классифицированы как ЗЕЛЕНЫЕ или КРАСНЫЕ. Наша задача - классифицировать новые случаи по мере их поступления, т. е. Решать, к какому классу они принадлежат, на основе существующих в настоящее время объектов.

Поскольку в два раза больше ЗЕЛЕНЫХ объектов, чем КРАСНЫХ, разумно полагать, что новый случай (который еще не был отмечен) в два раза чаще имеет членство ЗЕЛЕННЫЙ, а не КРАСНЫЙ. В байесовском анализе эта вера известна как предварительная вероятность. Предварительные вероятности основаны на предыдущем опыте, в данном случае проценте объектов ЗЕЛЕННЫЕ и КРАСНЫЕ и часто используются для прогнозирования результатов до того, как они действительно произойдут.

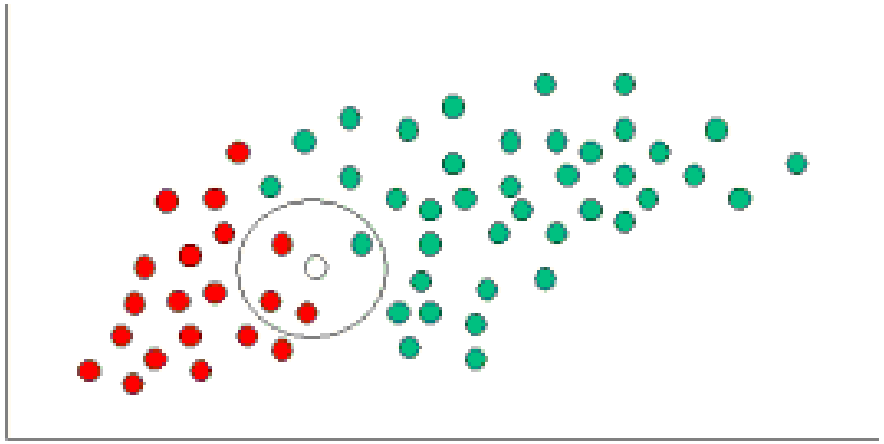
Оценка тональности. Naïve Bayes Classification

Таким образом, мы можем написать:

- Предварительная вероятность GREEN: количество ЗЕЛЕНЫХ объектов / общее количество объектов
- Предварительная вероятность RED: количество RED-объектов / общее количество объектов
- Поскольку в общей сложности 60 объектов, из которых 40 из них являются ЗЕЛЕНЫМИ и 20 КРАСНЫМИ, наши предыдущие вероятности для членства в классе:
- Предварительная вероятность для ЗЕЛЕНЫХ: 40/60
- Предварительная вероятность для КРАСНОГО: 20/60

Оценка тональности. Naïve Bayes Classification

Сформулировав нашу предварительную вероятность, мы теперь готовы классифицировать новый объект (круг WHITE на диаграмме ниже). Поскольку объекты хорошо сгруппированы, разумно предположить, что чем больше ЗЕЛЕНЫХ (или КРАСНЫХ) объектов в окрестности X, тем больше вероятность, что новые случаи относятся к этому конкретному цвету. Чтобы измерить эту вероятность, мы нарисуем круг вокруг X, который охватывает число (выбирается априори) точек независимо от их ярлыков класса. Затем мы вычисляем количество точек в круге, принадлежащих каждой метке класса. Из этого мы вычисляем вероятность:



Оценка тональности. Naïve Bayes Classification

$$Likelihood\ of\ X\ given\ GREEN \propto \frac{Number\ of\ GREEN\ in\ the\ vicinity\ of\ X}{Total\ number\ of\ GREEN\ cases}$$

$$Likelihood\ of\ X\ given\ RED \propto \frac{Number\ of\ RED\ in\ the\ vicinity\ of\ X}{Total\ number\ of\ RED\ cases}$$

Из приведенной выше иллюстрации ясно, что вероятность X, заданная ЗЕЛЕНЫМ, меньше, чем вероятность X, заданная RED, поскольку круг включает в себя 1 ЗЕЛЕНЫЙ объект и 3 КРАСНЫХ. Таким образом:

$$Probability\ of\ X\ given\ GREEN \propto \frac{1}{40} \qquad Probability\ of\ X\ given\ RED \propto \frac{3}{20}$$

Хотя предыдущие вероятности указывают на то, что X может принадлежать ЗЕЛЕНЫМ (учитывая, что в два раза больше ЗЕЛЕНЫХ по сравнению с КРАСНЫМ), вероятность указывает иначе; что член класса X является КРАСНЫМ (учитывая, что в окрестности X есть больше RED объектов, чем GREEN). В байесовском анализе окончательная классификация производится путем объединения обоих источников информации, то есть предшествующей и вероятности, для формирования задней вероятности с использованием так называемого правила Байеса (названного в честь преподобного Томаса Байеса 1702-1761).

Оценка тональности. Naïve Bayes Classification

$$\textit{Posterior probability of } X \textit{ being } GREEN \propto$$
$$\textit{Prior probability of } GREEN \times \textit{Likelihood of } X \textit{ given } GREEN$$
$$= \frac{4}{6} \times \frac{1}{40} = \frac{1}{60}$$
$$\textit{Posterior probability of } X \textit{ being } RED \propto$$
$$\textit{Prior probability of } RED \times \textit{Likelihood of } X \textit{ given } RED$$
$$= \frac{2}{6} \times \frac{3}{20} = \frac{1}{20}$$

Наконец, мы классифицируем X как КРАСНЫЙ, поскольку его членство в классе достигает наибольшей апостериорной вероятности.

Результаты

В результате работы системы было спарсено около 490000 новостей и 9,5 миллионов комментариев.

В качестве тренировочного и тестового набора для анализа тональности использовались более 200000 заранее размеченных твитов (комментариев). Этот твиты были разделены на два набора: тренировочный и тестовый.

```
train_tweets = tweets[0:75000] + tweets[100000:175000]
test_tweets = tweets[75000:100000] + tweets[175000:200000]
```

После того как классификатор был обучен на 150000 твитах и протестирован на 50000 твитах, был достигнута точность в 74%.

Для улучшения анализатора тональности был создан инструмент для оценки комментариев в ручном режиме. В нем экспертам даются комментарии, к которым они могут дать оценку: положительный, отрицательный или же нейтральный. Каждый комментарии оценивается тремя экспертами. В конце для каждого комментария берется среднее значение тональности. После того, как достаточное количество комментариев будет оценено, на первом этапе эти данные используются для проверки качества анализатора на реальных данных. Далее, используя эти же данные, анализатор до обучается и расширяет свою базу знании.

В ходе экспертной оценки было оценено более 15000 комментариев и достигнута точность в 47.73%. Алгоритм обученный на твиттерах показывает точность хуже, так как он не учитывает особенности комментирования казахстанских пользователей. Соответственно, текущий анализатор надо дообучить используя собранные данных от экспертных оценок.

Источники

- https://doc.scrapy.org/en/latest/intro/tutorial.html
- https://www.nltk.org/api/nltk.tokenize.html
- https://www.nltk.org/api/nltk.tokenize.html
- https://ru.wikipedia.org/wiki/XPath
- https://ru.wikipedia.org/wiki/XPath
- https://research-journal.org/languages/k-voprosu-o-tokenizacii-teksta/
- https://medium.com/@eigenein/%D1%81%D1%82%D0%B5%D0%BC%D0%BC%D0%B5%D1%80-%D0%BF%D0%BE%D1%80%D1%82%D0%B5%D1%80%D0%B0-%D0%B4%D0%BB%D1%8F-%D1%80%D1%83%D1%81%D1%81%D0%BA%D0%BE%D0%B3%D0%BE-%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0-d41c38b2d340

Спасибо за внимание!