//

Sentiment Analysis NLTK API

____

maXbox Starter 82_2 - How to make a Sentiment Analysis, Max Kleiner

"Yesterday I was clever, so I wanted to change the world. Today I am wise, so I am changing myself." - Rumi

As you way know, we went through the last magazine report on the BBC News feed, line by line; now we want to equip and enlarge these texts with a sentiment analysis like the following:

11: French MP and billionaire Olivier Dassault dies in helicopter crash:
Sun, 07 Mar 2021 20:04:17 GMT

```
{"probability": {"neg": 0.46705201547833042, "neutral": 0.81510771060379195, "pos": 0.53294798452166958}, "label": "neutral"}
```

So the label shows neutral in the middle of possibilities. The english sentiment uses classifiers trained on both twitter sentiment as well as movie reviews from the data sets. The dutch and french sentiment is based on book reviews.

Because of the nature of the text and its categories, the classification we will be doing is a form of sentiment analysis or opinion mining. If the classifier returns pos, then the text expresses a positive sentiment, whereas if we get neg, then the text expresses a negative sentiment.

pic: http://www.softwareschule.ch/examples/starter82_2_sentimentweb.png

This procedure of discovering and classifying opinions expressed in a piece of text (like comments/feedback text/news feed in our case) is called the sentiment analysis. The intended output of this analysis would be to determine whether the producers mindset toward a topic, product, headline or service etc., is in most cases neutral, positive, or negative.

Lets get started with the use of an API:

```
Const
  URLSentimentAPI2='http://text-processing.com/api/sentiment/';
```

Our text lines are based on the BBC-News Feed:

```
  BBCFeed = 'http://feeds.bbci.co.uk/news/world/rss.xml';
```

To call the API we use a late binding OLE Automation from <msxml2.xmlhttp>. The text-processing.com API is a simple JSON over HTTP web service for text mining and natural language processing. It is currently free and open for public use without authentication or login, but could be changed in the future. As of JSON we use the delphi4json library to parse the return.

The script you can find at:
http://www.softwareschule.ch/examples/newssentiment2.txt

```
  XMLhttp:= CreateOleObject('msxml2.xmlhttp')
  XMLhttp.Open('POST', URLSentimentAPI2, False)   //False: async
  XMLhttp.setRequestHeader('Content-Type','application/json');
  XMLhttp.Send('text='+'"'+textin+'"'+CRLF+'language=english');
  response:=  XMLhttp.responseText; //assign data
  writeln(response)
  writeln('statuscode: '+itoa(XMLhttp.status;))
```

On success, a 200 OK response will be returned containing a JSON object that looks like:

```
  {"probability":
    {"neg": 0.375174845959971884,
     "neutral": 0.0910342745413776914,
     "pos": 0.62482515404028116},
     "label": "pos"}
```

A 503 Throttled response will be returned if you exceed the daily request limit. Using async = false in Open() is not always recommended, but for a few small requests this can be ok. Remember that the script will NOT continue to execute, until the server response is ready. If the server is busy or slow, the application will hang or stop.

Anyway we open our XMLhttpObject (which is late binding) as not asynchronous, that is to say, synchronous because we just post a single block of data with send().

The send() method (XMLhttp.Send();) needs more explanation: send() accepts an optional parameter which lets you specify the requests body; this is primarily used for requests such as PUT or POST. If the request method is GET or HEAD, the body parameter is ignored and the request body is set to null. Im not sure if the content-type is the right (text or application), the MIME media type for JSON text is application/json and the default encoding is UTF-8. (Source: RFC 4627).

{content-type: application/json; charset=utf-8}

To analyze some text, we do an HTTP POST to our API with form encoded data containg the text we want to analyze. We get back a JSON object response with 2 attributes label and probability which we parse with a JSON object (more of that in the next number):

```
  with TJson.create() do begin
    clear;
    parse(response);
    cnode:= JsonObject.items[0].name;  //'probability'
    writeln(itoa(JsonObject.count));
    writeln('prob: '+values[cnode].asObject.values['neutral'].asstring);
    writeln('result: '+(values['label'].asstring));
    free;
  end;
```

```
>>> 2
>>> prob: 0.854074272795421
>>> result: neutral
```

As you may see many of the most commonly used words pr phrases are insignificant when it comes to discerning the meaning of a phrase. For example, in the phrase the movie was terrible, the most significant words are movie and terrible, while the and was are almost useless. You could get the same meaning if you took them out, that is, movie terrible or terrible movie. Either way, the sentiment is the same.

Another approach is to measure the sentiment of face feelings with 3 flavours: Joy, Sadness and Anger or Disbelief, but thats kind of research.

Our quote from above results in:

Sentiment of: "Yesterday I was clever, so I wanted to change the world. Today I am wise, so I am changing myself".

{"probability": {"neg": 0.375174845959971884, "neutral": 0.0910342745413776914, "pos": 0.62482515404028116}, "label": "pos"}

statuscode: 200

pic: http://www.softwareschule.ch/examples/starter82_2_sentimentweb2.png

The NLTK (Natural Language Toolkit) is a leading platform for building Python or API programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet and many others.

Conclusion:

This is a demonstration of sentiment analysis using a NLTK 2.0.4 powered text classification process with msxml2.xmlhttp and TJson objects. It can tell you whether it thinks the text you enter below expresses positive sentiment, negative sentiment, or if it is neutral.

The feedback results will be more accurate on text that is similar to original training data. If you get an odd result, it could be the words you have used are unrecognized. Try entering more words or blocks to improve accuracy. Note that the public API is for non-commercial purposes, and each method is throttled to 1000 calls per day per IP.

You may also test the same context with different languages, the default language is english, but this API also supports dutch and french, but dont forget to change the language in the API call:

```
XMLhttp.Send('text='+'"'+textin+'"'+CRLF+'language=dutch');
```

```
Trump's false election fraud claims face a dead end
{"probability": {"neg": 0.52, "neutral": 0.64, "pos": 0.47}, "label": "neutral"}
Trumps falsche Wahlbetrugsansprüche stehen vor einer Sackgasse
Trump's valse verkiezingsfraudeclaims lopen dood
Les fausses allégations de fraude électorale de Trump font face à une impasse
{"probability": {"neg": 0.33, "neutral": 0.46, "pos": 0.66}, "label": "pos"}
Trump's valse verkiezingsfraudeclaims lopen dood
{"probability": {"neg": 0.48, "neutral": 0.72, "pos": 0.51}, "label": "neutral"}
```

Ref:
    http://www.softwareschule.ch/examples/newssentiment2.txt
    http://text-processing.com/docs/sentiment.html
    https://www.nltk.org/
    script: 1017_RSSDatacampSSLStreamSentiment2.pas
Doc:
    https://maxbox4.wordpress.com

Appendix: Alternate HTTPPost-Routine:

```
function GetBlogStream8Sentiment(const S_API, pData: string;
                                       astrm: TStringStream): TStringStream;
begin
  sr:='text='+HTTPEncode(pData)+CRLF;
  sr:= sr+'language=english';
  if HttpPostURL(S_API, sr, astrm) then
    result:= astrm;
end;
```