2.2. Preprocessing_77k_tweets

October 19, 2017

0.1 Load the data

In this step we read the data in a json format from the file and we store the tweet text info in a pandas DataFrame

```
In [ ]: import json
        import pandas as pd

        tweets_file = 'websummit_dump_20151106155110'
        with open(tweets_file) as f:
            tweets = json.load(f)

        print('# of tweets:', len(tweets))

        tweet_text = [tweet['text'] for tweet in tweets]
        df = pd.DataFrame({'text': tweet_text})
        df.head()
```

0.2 Define the tweet preprocessor

We define a preprocessor withch finds the diferent elements of the tweet using regular expresions and it identifies them.

```
In [2]: import re

        class TweetPreprocessor(object):

            def __init__(self):
                self.FLAGS = re.MULTILINE | re.DOTALL
                self.ALLCAPS = '<allcaps>'
                self.HASHTAG = '<hashtag>'
                self.URL = '<url>'
                self.USER = '<user>'
                self.SMILE = '<smile>'
                self.LOLFACE = '<lolface>'
                self.SADFACE = '<sadface>'
                self.NEUTRALFACE = '<neutralface>'
                self.HEART = '<heart>'
                self.NUMBER = '<number>'
                self.REPEAT = '<repeat>'
                self.ELONG = '<elong>'

            def _hashtag(self, text):
                text = text.group()
                hashtag_body = text[1:]
                if hashtag_body.isupper():
                    result = (self.HASHTAG + " {} " + self.ALLCAPS).format(hashtag_body)
                else:
                    result = " ".join([self.HASHTAG] + re.findall(r"(?=[A-Z])", hashtag_body, 
                return result

            def _allcaps(self, text):
                text = text.group()
                return text.lower() + ' ' + self.ALLCAPS

            def preprocess(self, text):
                eyes, nose = r"[8:=;]", r"['`\-]?"

                re_sub = lambda pattern, repl: re.sub(pattern, repl, text, flags=self.FLAGS)

                text = re_sub(r"https?:\/\/\S+\b|www\.(\w+\.)+\S*", self.URL)
                text = re_sub(r"/"," / ")
                text = re_sub(r"@\w+", self.USER)
                text = re_sub(r"{}{}[)dD]+|[)dD]+{}{}".format(eyes, nose, nose, eyes), self.SM
                text = re_sub(r"{}{}p+".format(eyes, nose), self.LOLFACE)
                text = re_sub(r"{}{}\(+|\)+{}{}".format(eyes, nose, nose, eyes), self.SADFACE)
                text = re_sub(r"{}{}[\/|l*]".format(eyes, nose), self.NEUTRALFACE)
                text = re_sub(r"<3", self.HEART)
                text = re_sub(r"[-+]?[.\d]*[\d]+[:,.\d]*", self.NUMBER)
                text = re_sub(r"#\S+", self._hashtag)
                text = re_sub(r"([!?.]){2,}", r"\1 " + self.REPEAT)
                text = re_sub(r"\b(\S*?)(.)\2{2,}\b", r"\1\2 " + self.ELONG)

                text = re_sub(r"([A-Z]){2,}", self._allcaps)

                return text.lower()
```

```
In [3]: tweet_processor = TweetPreprocessor()

        # an example:
        tweet = "@sarahtavel What #MustHave #tech gadget can you not travel without? Stop by s
        print("Before: " + tweet + "\n")
        print("After: " + tweet_processor.preprocess(tweet))
```

```
Before: @sarahtavel What #MustHave #tech gadget can you not travel without? Stop by stand D131
```

```
After: <user> what <hashtag>   <hashtag> gadget can you not travel without? stop by stand d<nu
```

```
In [4]: import nltk
        from nltk.corpus import stopwords
        from nltk.tokenize import TweetTokenizer

        # We define a tweet tokenizer withc split the text in words
        tknzr = TweetTokenizer()

        # We define the stopwords that are words like 'and', 'or', 'not' that do not give rele
        stop = stopwords.words('english')

        # Add the tags defined in the preprocessing to the stepwords
        stop += ['<hashtag>', '<url>', '<allcaps>', '<number>', '<user>', '<repeat>', '<elong>

        df['text_processed'] = ""
        index = 0

        for tweet in df['text']:
            # Remove the no relevant information from the tweets
            parts = tknzr.tokenize(tweet_processor.preprocess(tweet))
            clean = [i for i in parts if i not in stop]
            df['text_processed'][index] = clean
            index += 1
```

```
[nltk_data] Downloading package stopwords to /home/set92/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
In [5]: df['text_processed'].size
```

```
Out[5]: 77111
```

```
In [6]: df.to_pickle('77k_df.pickle')
```