Report Of Tokenizer:

The aim of assignment was to break tweets into suitable tokens using regex expressions for various cases.

Regex with their use:::

1.**URL**
(https?:\/\/)?([\w\-]+\.)?[\w\-]+\.[\w]{2,6}([\/\?][\w\-\!\?\%\~\&\=\+]+[\.]?[\w\-\!\?\%\~\&\=\+]*)*\/?

2.**HYPHEN**
[\w]+\-[\w]+

3.**ABBREVIATION**
([A-Z][\s]?[\.]?[\s]){3,8}

4.**USER_REFERENCE**
@([A-Za-z0-9_]+\:)

** It handles both the “@writer:” and “@mention” as different tokens.

5.**EMOTICON**
([\<\>\}]?[\:\;B][\'\"]?[\-\^]?[\)\(\#\@\$SPpdDOoL\|\/\\]{1,3})|([\)\(\#\@\$dOo\|\/\\]{1,3}[\-\^]?[\'\"]?[\:][\<\>\{E]?)"

6.**HASHTAG**
(?:\#+[\w_]+[\w\'_\-]*[\w_]+)

Algorithm:::

1. First the entire tweet data is loaded and read, then the individual tweets are separated using delimiter “\n”.
2. Then using our script each and every tweet is tokenized separately.
3. Keeping all the special cases to be handled in the regex expression and then later comparing the regex to check the presence of any of the pattern.
4. Here “preprocess(tweet)” returns list of tuples which in themselves contain multiple spaces also.
5. Avoid the list in which only tuples present.
6. return and write the list of tokes of the tweet in the file.

Cases Not handled:::
1. Dates like 28th-Aug-1994, 24th Aug'94 are not handled.
2. Words like New York are tokenized as “New” and “York”, not like “New York”.
3. cases like can't , don't , you'll are not handled.