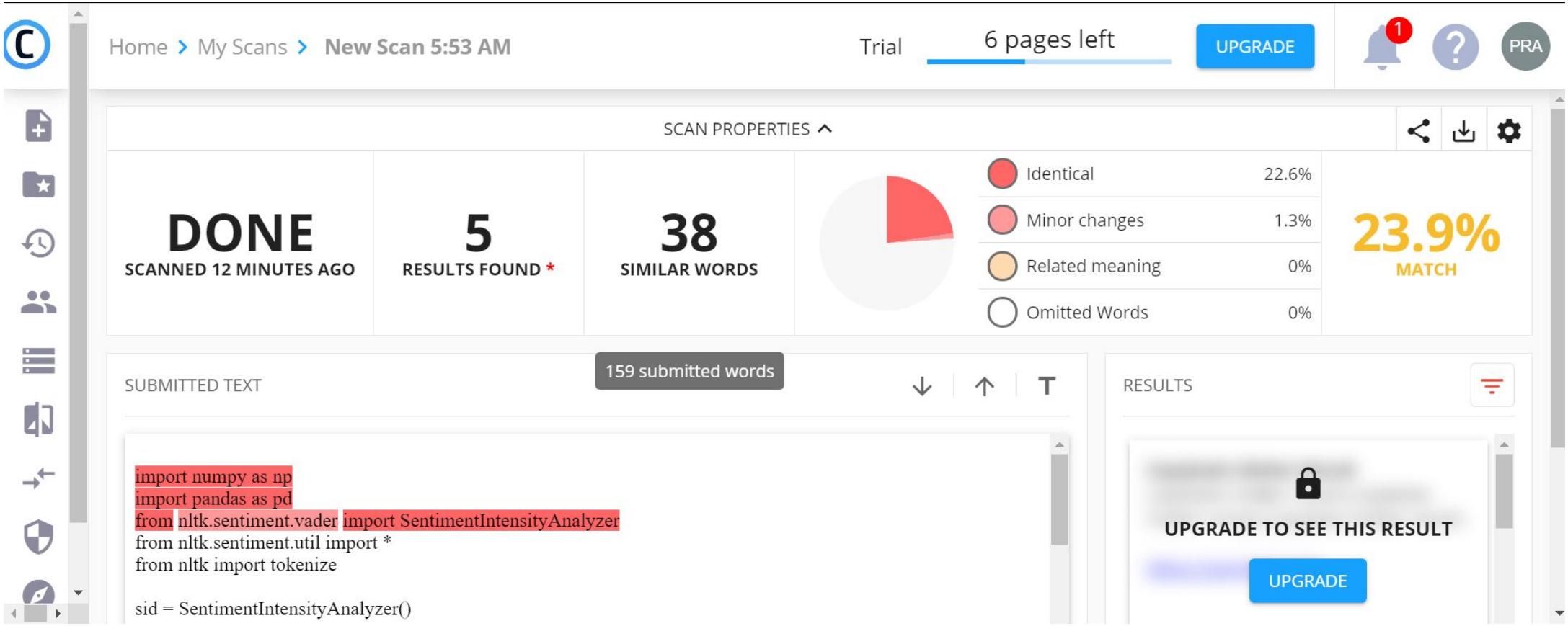
Home > My Scans > New Scan 5:53 AM
Trial
6 pages left
UPGRADE
PRA
SCAN PROPERTIES
DONE
SCANNED 12 MINUTES AGO
5
RESULTS FOUND *
38
SIMILAR WORDS
Identical 22.6%
Minor changes 1.3%
Related meaning 0%
Omitted Words 0%
23.9%
MATCH
SUBMITTED TEXT
159 submitted words
RESULTS
import numpy as np
import pandas as pd
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.sentiment.util import *
from nltk import tokenize
sid = SentimentIntensityAnalyzer()
UPGRADE TO SEE THIS RESULT
UPGRADE

SUBMITTED TEXT

```
sid = SentimentIntensityAnalyzer()

tweets = pd.read_csv('narendramodi1.csv')

def remove_pattern(input_txt, pattern):
r = re.findall(pattern, input_txt)
for word in r:
input_txt = re.sub(word, "", input_txt)
return input_txt

df['clean_tweet'] = np.vectorize(remove_pattern)(df['Tweets'], "@[\w]*")

df['clean_tweet'] = df['clean_tweet'].str.replace("[^a-zA-Z#]", " ")
df['clean_tweet'] = df['clean_tweet'].apply(lambda x: " ".join([w for w in x.split() if len(w)>3]))

tokenized_tweet = df['clean_tweet'].apply(lambda x: x.split())
tokenized_tweet.head()
```

SUBMITTED TEXT

```
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

tokenized_tweet = tokenized_tweet.apply(lambda sentence: [stemmer.stem(word) for word in sentence])

for i in range(len(tokenized_tweet)):
tokenized_tweet[i] = " ".join(tokenized_tweet[i])
df['clean_tweet'] = tokenized_tweet

tweets['sentiment_compound_polarity']=tweets.Tweets.apply(lambda x:sid.polarity_scores(x)['compound'])
tweets['sentiment_neutral']=tweets.Tweets.apply(lambda x:sid.polarity_scores(x)['neu'])
tweets['sentiment_negative']=tweets.Tweets.apply(lambda x:sid.polarity_scores(x)['neg'])
tweets['sentiment_pos']=tweets.Tweets.apply(lambda x:sid.polarity_scores(x)['pos'])
tweets['sentiment_type']=' '

tweets.loc[tweets.sentiment_compound_polarity>0,'sentiment_type']='POSITIVE'
tweets.loc[tweets.sentiment_compound_polarity==0,'sentiment_type']='NEUTRAL'
tweets.loc[tweets.sentiment_compound_polarity<0,'sentiment_type']='NEGATIVE'
```

SUBMITTED TEXT

```
tweets.sentiment_type.value_counts().plot(kind='bar',title="sentiment analysis")


import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
from subprocess import check_output
from wordcloud import WordCloud, STOPWORDS

mpl.rcParams['font.size']=12
mpl.rcParams['savefig.dpi']=100
mpl.rcParams['figure.subplot.bottom']=.1
stopwords = set(STOPWORDS)
data = pd.read_csv("narendramodi1.csv")
wordcloud = WordCloud(
background_color='white',
stopwords=stopwords,
max_words=100,
```

SUBMITTED TEXT

```
mpl.rcParams['savefig.dpi']=100
mpl.rcParams['figure.subplot.bottom']=.1
stopwords = set(STOPWORDS)
data = pd.read_csv("narendramodi1.csv")
wordcloud = WordCloud(
background_color='white',
stopwords=stopwords,
max_words=100,
max_font_size=40,
random_state=42
).generate(str(data['Tweets']))
print(wordcloud)

fig = plt.figure(1)
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
fig.savefig("word1.png", dpi=900)
```