

Reddit Analysis Communities Only

March 30, 2016

```
In [22]: %matplotlib inline
```

```
import cPickle as pickle
import matplotlib.pyplot as plt
plt.style.use('ggplot')
%matplotlib inline
%config InlineBackend.figure_format = 'svg'

import logging, os, sys

logger = logging.getLogger('root')

program = os.path.basename(sys.argv[0])
logger = logging.getLogger(program)
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s')
logging.root.setLevel(level=logging.INFO)
logger.info("running %s" % ' '.join(sys.argv))

import numpy

import pickle as cPickle

import pandas
```

```
INFO:__main__.py:running /Users/linanqiu/.virtualenv/default/lib/python2.7/site-packages/ipykernel/__main__
```

```
In [23]: def plot_metareddit(metareddit):
    reddit_substitute_key = pickle.load(open('substitute_keys/reddit_%s_substitute_key.pkl' % m
    vocabs_loaded = pickle.load(open('reddit_vocabs_reference/vocabs_reference_reddit_%s_commun

    reddit_substitute_key_inv = {v: k for k, v in reddit_substitute_key.items()}

    for word, counts in vocabs_loaded.iteritems():
        vocabs_loaded[word]['is_codeword'] = word in reddit_substitute_key_inv

    reddit_counts_codewords = [counts['count'] for word, counts in vocabs_loaded.iteritems() i
    reddit_counts_usual = [counts['count'] for word, counts in vocabs_loaded.iteritems() if no

    f, axes = plt.subplots(2, sharex=True)
    f.suptitle('Reddit %s' % metareddit)
    axes[0].set_title('Non-Codewords')
    axes[1].set_title('Codewords')
    axes[0].set_ylabel('Word Count')
    axes[1].set_ylabel('Word Count')
```

```

bins = numpy.linspace(0, 300, 300)
axes[0].hist(reddit_counts_usual, bins)
axes[1].hist(reddit_counts_codewords, bins)
plt.savefig('reddit_figures/reddit_%s.pdf' % metareddit)
plt.show()

def summary_metareddit(metareddit):

    reddit_substitute_key = pickle.load(open('substitute_keys/reddit_%s_substitute_key.pkl' % metareddit, 'rb'))
    vocabs_loaded = pickle.load(open('reddit_vocabs_reference/vocabs_reference_reddit_%s_communities.pkl' % metareddit, 'rb'))

    reddit_substitute_key_inv = {v: k for k, v in reddit_substitute_key.items()}

    for word, counts in vocabs_loaded.iteritems():
        vocabs_loaded[word]['is_codeword'] = word in reddit_substitute_key_inv

    words = [{ 'word': word, 'count': vocabs_loaded[word]['count'], 'is_codeword': vocabs_loaded[word]['is_codeword'] } for word in vocabs_loaded.keys()]

    words = sorted(words, key=lambda k: k['count'])

    total_codeword_count = len(reddit_substitute_key)

    true_positive = 0 # selected as codeword and is codeword
    true_negative = len(words) - len(reddit_substitute_key) # not selected as codeword and is not codeword
    false_positive = 0 # selected as codeword and isn't codeword
    false_negative = len(reddit_substitute_key) # not selected as codeword and is codeword

    precision_recall_f1s = []

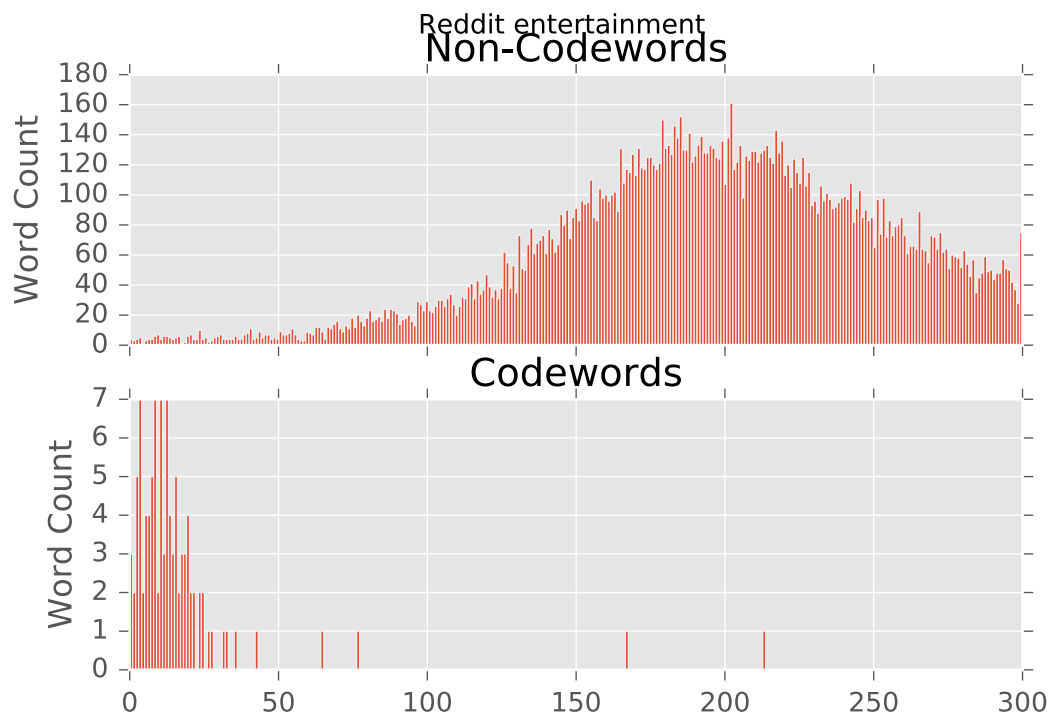
    for word_count in range(0, len(words)):
        word_tuple = words[word_count]
        if word_tuple['is_codeword']:
            true_positive += 1
            false_negative -= 1
        else:
            false_positive += 1
            true_negative -= 1
        precision = float(true_positive) / float(true_positive + false_positive)
        recall = float(true_positive) / float(true_positive + false_negative)
        try:
            f1 = precision * recall / (precision + recall)
        except ZeroDivisionError:
            f1 = 0

        precision_recall_f1s.append({'precision': precision, 'recall': recall, 'f1': f1})

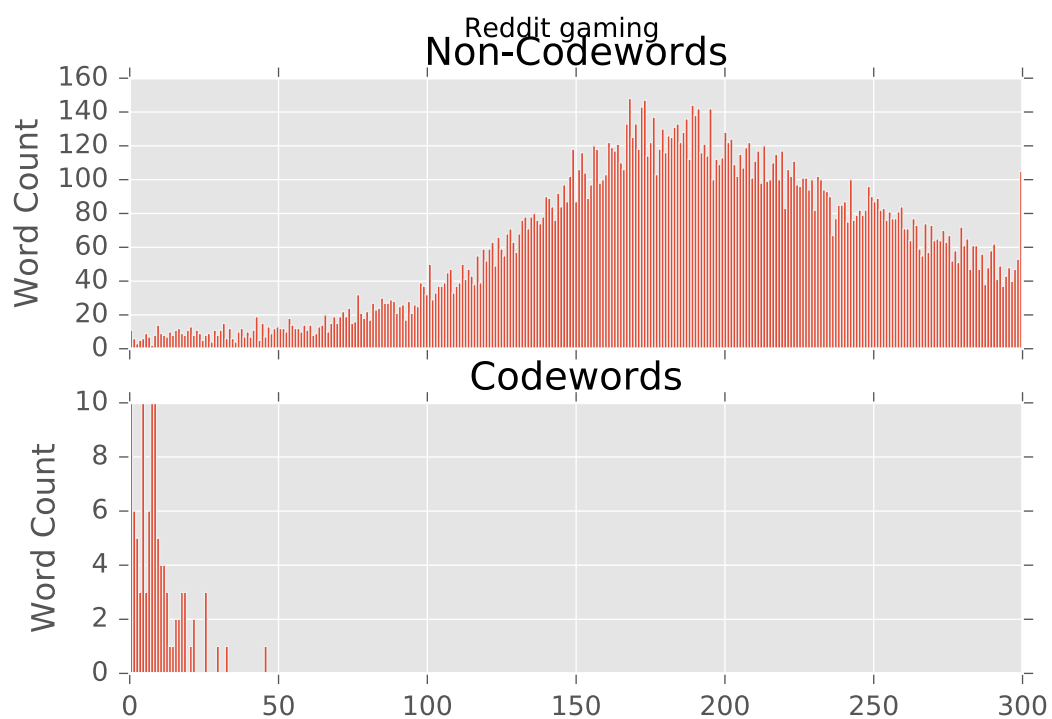
    return precision_recall_f1s

In [24]: plot_metareddit('entertainment')

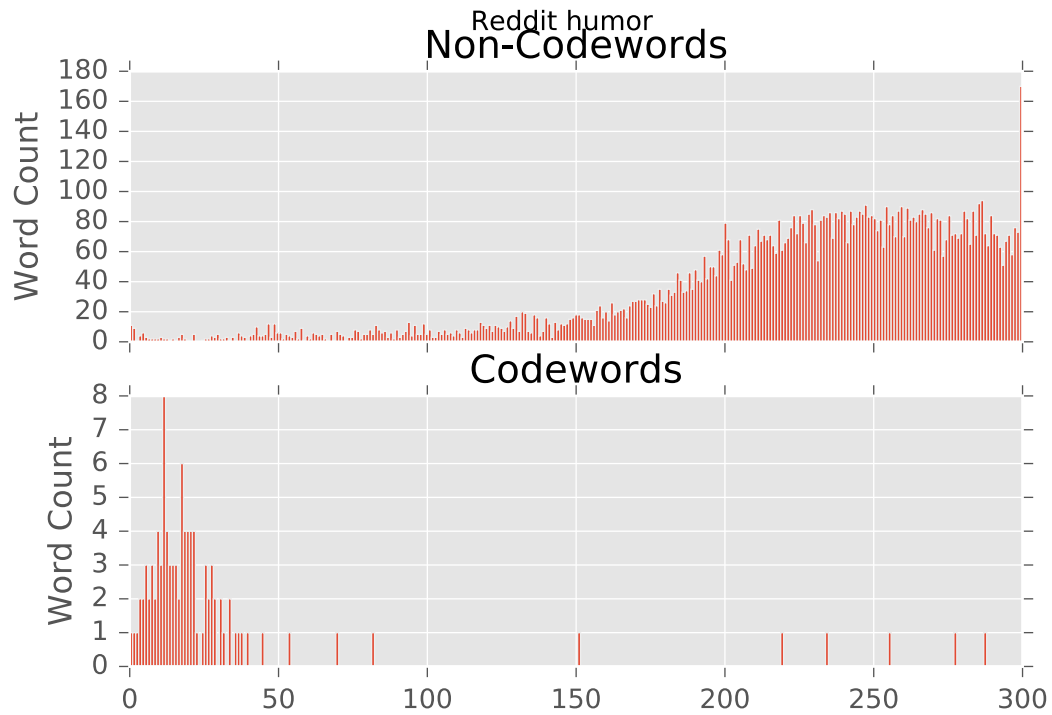
```



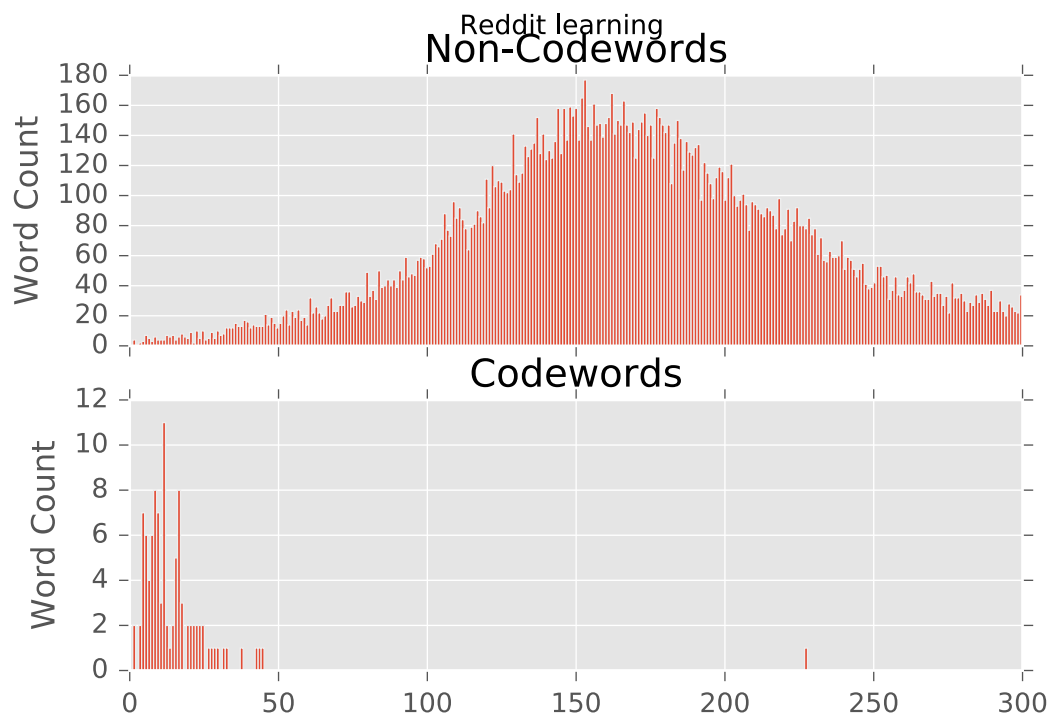
```
In [25]: plot_metareddit('gaming')
```



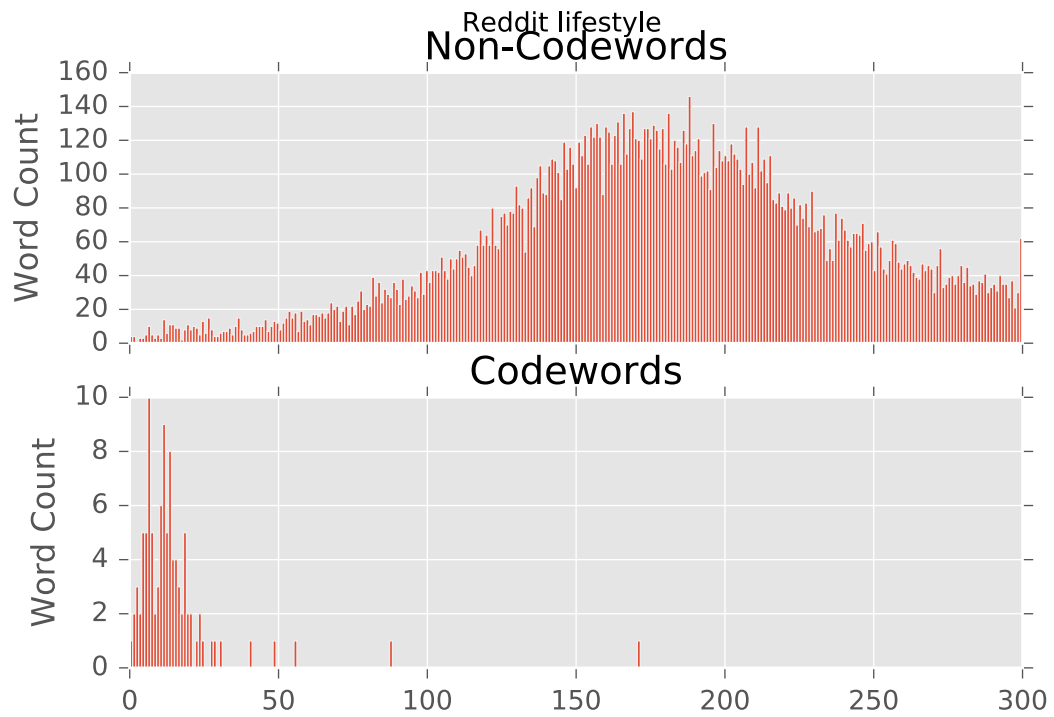
```
In [26]: plot_metareddit('humor')
```



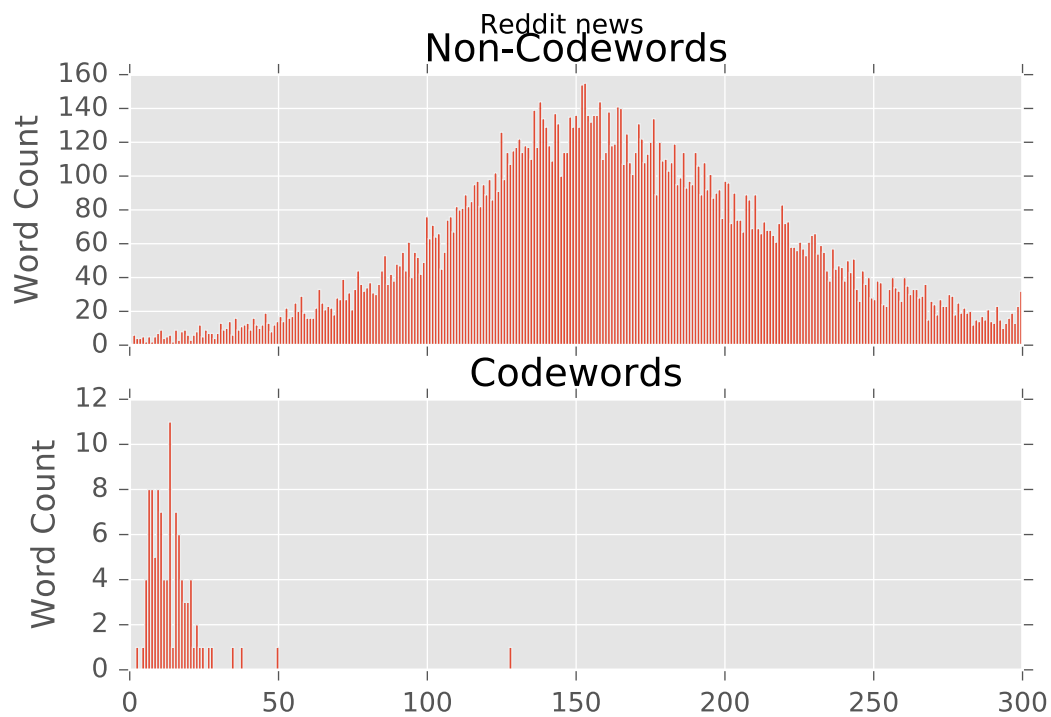
```
In [27]: plot_metareddit('learning')
```



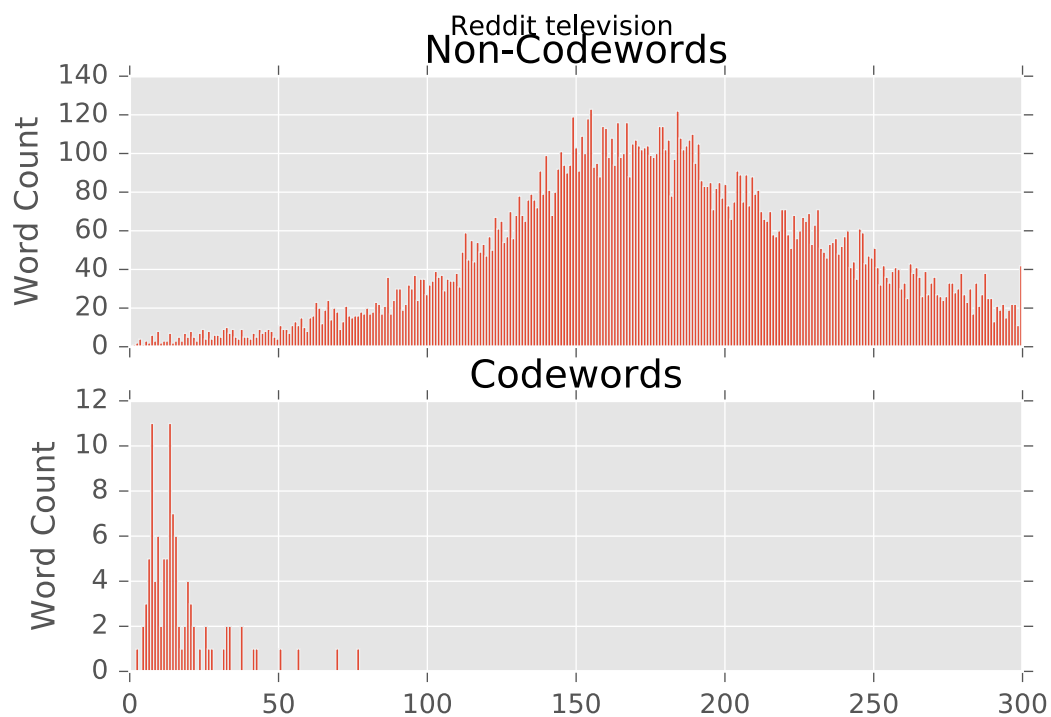
```
In [28]: plot_metareddit('lifestyle')
```



```
In [29]: plot_metareddit('news')
```

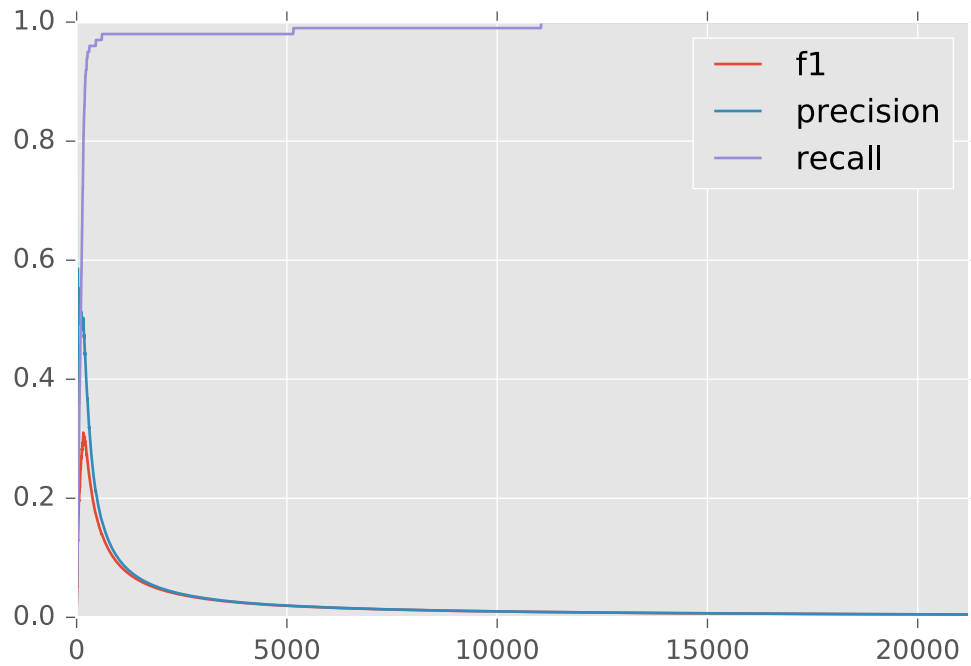


```
In [30]: plot_metareddit('television')
```



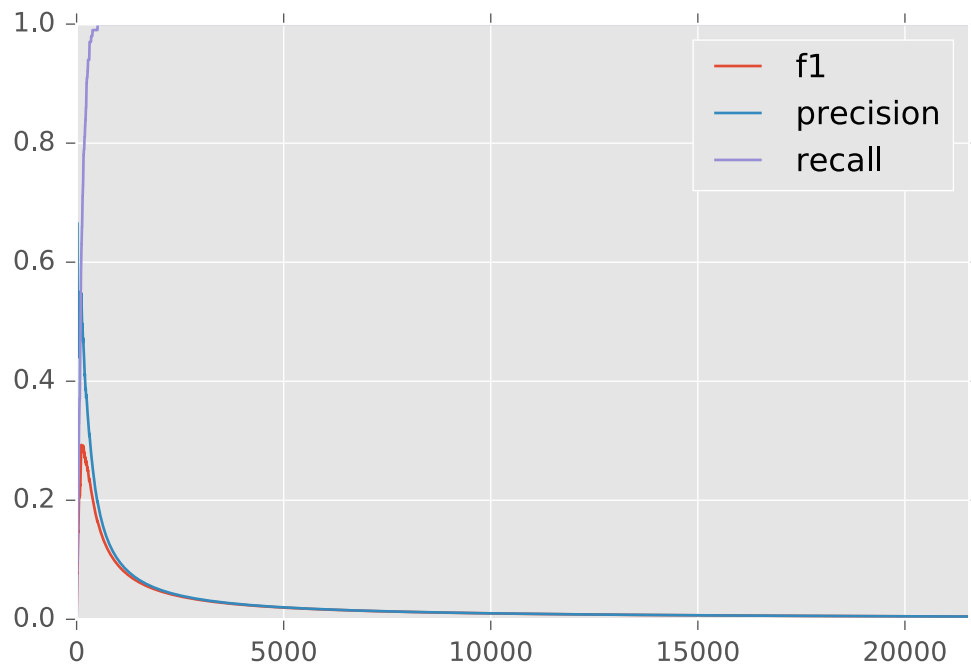
```
In [31]: summary_entertainment = pandas.DataFrame(summary_metareddit('entertainment'))
summary_entertainment.plot()
```

```
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x11172a050>
```



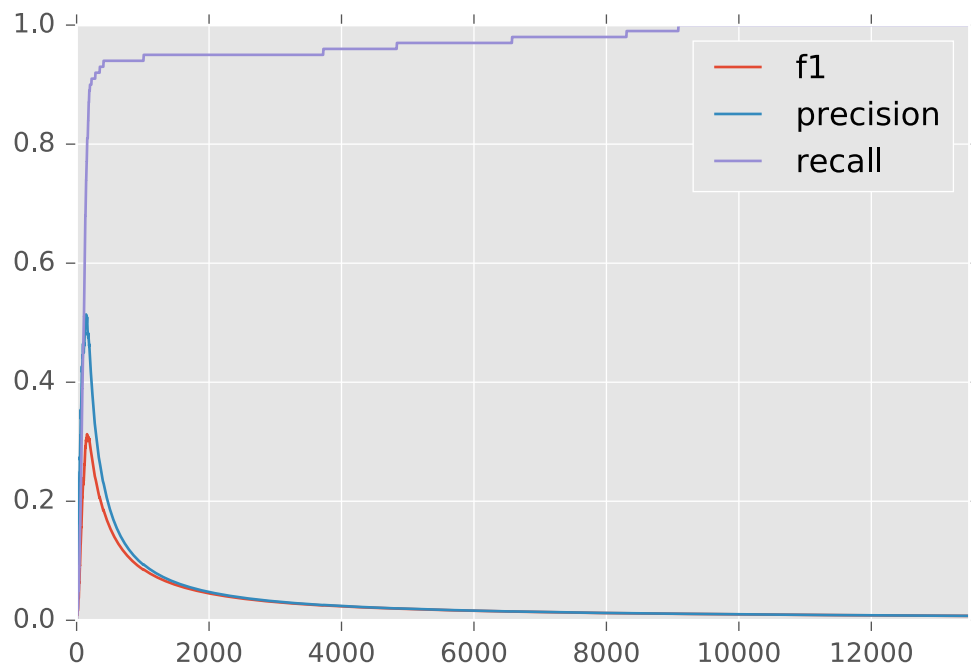
```
In [32]: summary_gaming = pandas.DataFrame(summary_metareddit('gaming'))
summary_gaming.plot()
```

```
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x10cd1b810>
```



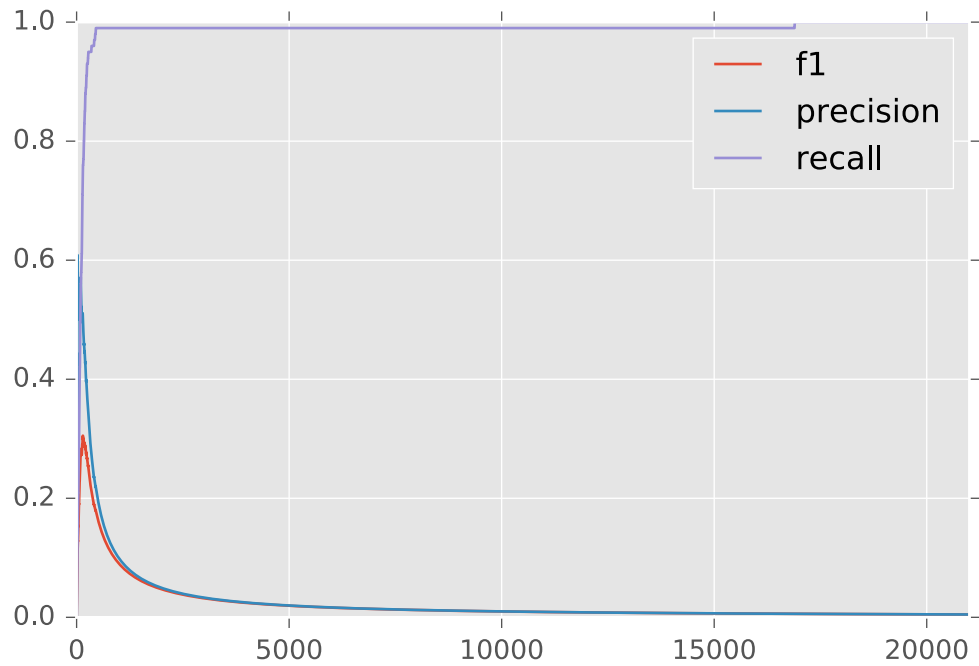
```
In [33]: summary_humor = pandas.DataFrame(summary_metareddit('humor'))
summary_humor.plot()
```

```
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x10c1458d0>
```



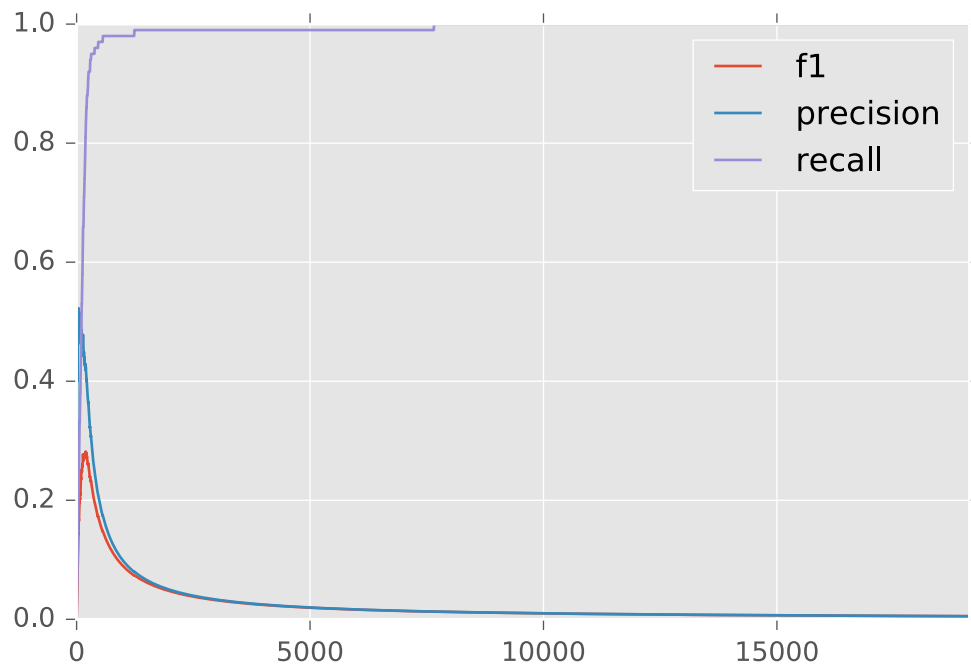

```
In [34]: summary_learning = pandas.DataFrame(summary_metareddit('learning'))
summary_learning.plot()
```

```
Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x10d0e0890>
```



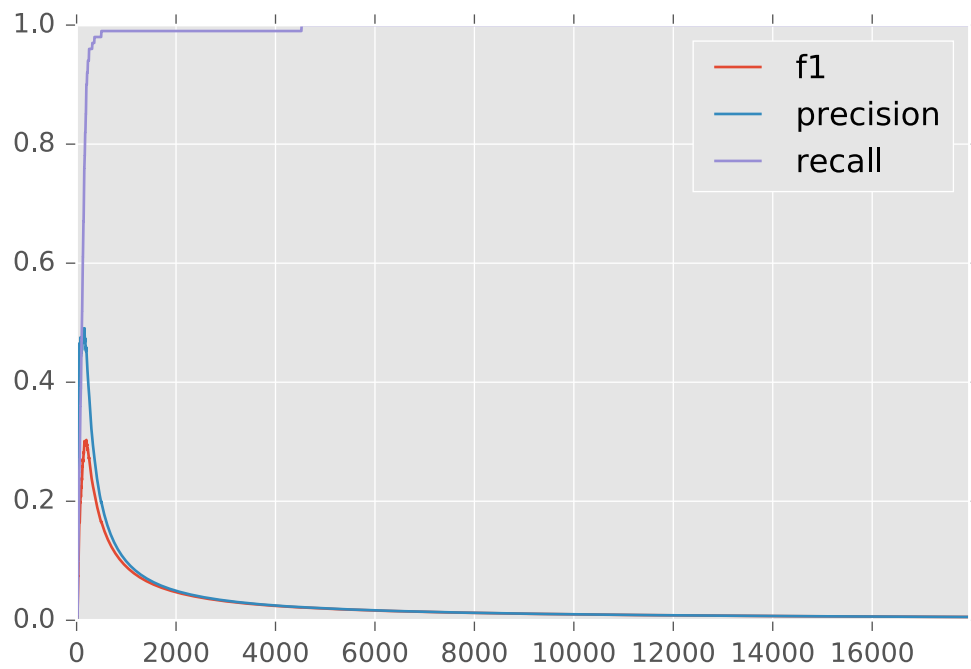
```
In [35]: summary_lifestyle = pandas.DataFrame(summary_metareddit('lifestyle'))
summary_lifestyle.plot()
```

```
Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x10c2a3fd0>
```



```
In [36]: summary_news = pandas.DataFrame(summary_metareddit('news'))  
summary_news.plot()
```

```
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x10aa80ad0>
```



```
In [37]: summary_television = pandas.DataFrame(summary_metareddit('television'))
summary_television.plot()
```

```
Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x10b5f88d0>
```

